PTO 06-5110

CY=JA DATE=19970516 KIND=A
PN=09-128276

COMPUTER SYSTEM AND FILE MANAGING METHOD USED BY THE SAME
[Keisanki shisutemu oyobi sono shisuemu de shiyo sareru fairu kanri hoho]

Toru Imai, et al.

UNITED STATES PATENT AND TRADEMARK OFFICE
Washington, D.C. July 2006

Translated by: FLS, Inc.

PUBLICATION COUNTRY	(19):	JP
DOCUMENT NUMBER	(11):	09128276
DOCUMENT KIND	(12):	A
PUBLICATION DATE	(43):	19970516
PUBLICATION DATE	(45):	
APPLICATION NUMBER	(21):	08191845
APPLICATION DATE	(22):	19960722
ADDITION TO	(61):	
INTERNATIONAL CLASSIFICATION	(51):	G06F 12/00
DOMESTIC CLASSIFICATION	(52):	
PRIORITY COUNTRY	(33):	JP
PRIORITY NUMBER	(31):	07219006
PRIORITY DATE	(32):	19950828
INVENTOR	(72):	IMAI, TORU; YOSHIDA, HIDEKI
APPLICANT	(71):	TOSHIBA CORPORATION
TITLE	(54):	COMPUTER SYSTEM AND FILE MANAGING METHOD USED BY THE SAME
FOREIGN TITLE	[54A]:	KEISANKI SHISUTEMU OYOBI SONO SHISUEMU DE SHIYO SARERU FAIRU KANRI HOHO

SPECIFICATION /1*

TITLE OF THE INVENTION

COMPUTER SYSTEM AND FILE MANAGIGN METHOD USED BY THE SAME

CLAIMS /2

1. A computer system for managing input and output of files to and from a storage medium capable of storing a plurality of files, characterized by providing judgment means for judging the necessity for erasure or compression of a file on said storage medium based on the free space on said storage medium, and

automatic file erasure/compression means for automatically selecting a file to be erased or compressed from among the files stored on said storage medium and following a procedure for erasing or compressing this file if these judgment means have judged file erasure or compression necessary.

2. A computer system according to Claim 1, characterized by said judgment means executing the judgment processing each time that a program running on said computer system issues a file writing request or at regular intervals.

3. A computer system for managing input and output of files to and from a storage medium capable of storing a plurality of files, characterized by providing judgment means for judging the necessity for erasure or compression of a file on said storage medium based on

*Numbers in the margin indicate pagination in the foreign text.

the free space on said storage medium,

automatic file erasure/compression means for selecting a file to be erased or compressed from among the files stored on said storage medium according to the priority of these files if these judgment means have judged file erasure or compression necessary, and

file erasure/compression means for following a procedure for erasing or compressing the file selected by these file erasure/compression means.

4. A computer system according to Claim 3, characterized by prescribing said priority of files by priority information predetermined by the user for each file, and

said file erasure/compression means referring to said priority information determined by the user to select a file to be erased or compressed.

5. A computer system according to Claim 3, characterized by prescribing said priority of files by the input/output history information for each file on said storage medium, and

said file erasure/compression means referring to said input/output history information to select a file to be erased or compressed.

6. A computer system for storing and managing a plurality of files using a storage medium,

characterized by providing means for storing file reading history information indicating whether each file stored on said storage medium has been read from said storage medium, how many times, and

the last date and time read,

file erasure/compression selection means for referring to said file reading history information to select a file to be erased or compressed from among the files stored on said storage medium, and

file erasure/compression means for following a procedure for erasing or compressing the file selected by these file erasure/compression means.

7. A computer system for managing input and output of files to and from a storage medium capable of storing a plurality of files, characterized by providing means for storing file size information for each file stored on said storage medium,

file erasure/compression selection means for referring to said file size information to select a file to be erased or compressed from among the files stored on said storage medium, and

file erasure/compression means for following a procedure for erasing or compressing the file selected by these file erasure/compression means.

8. A computer system for managing input and output of files to and from a storage medium capable of storing a plurality of files, characterized by providing means for storing file correlation information indicating correlations between file pairs with interchangeable content among the files stored on said storage medium,

file erasure/compression selection means for referring to said file

correlation information to select a file which is recoverable from the content of another file from among the files stored on said storage medium as a file to be erased or compressed, and

file erasure/compression means for following a procedure for erasing or compressing the file selected by these file erasure/compression means.

9. A computer system for managing input and output of files to and from a storage medium capable of storing a plurality of files,

characterized by providing means for detecting the number of references by other files for each file stored on said storage medium,

file erasure/compression selection means for selecting a file to be erased or compressed from among the files stored on said storage medium according to this detected number of references, and

file erasure/compression means for following a procedure for erasing or compressing the file selected by these file erasure/compression means.

10. A file managing method for managing the free space on a storage medium of a computer system,

characterized by judging the necessity for erasure or compression of a file on said storage medium based on the free space on said storage medium, /3

automatically selecting a file to be erased or compressed from among the files stored on said storage medium and following a procedure for

erasing or compressing this file if these judgment means have judged file erasure or compression necessary, and

being capable of automatically assuring free space on said storage medium.

11. A storage medium run on a computer system and storing a file management program for managing the free space of the file storage medium of this computer system,

characterized by said file management program

judging the necessity for erasure or compression of a file on said storage medium based on the free space on said storage medium, and

automatically selecting a file to be erased or compressed from among the files stored on said storage medium and following a procedure for erasing or compressing this file if these judgment means have judged file erasure or compression necessary.

12. A computer system for storing and managing a plurality of files using a storage medium,

characterized by providing file erasure means for selecting a file to be erased from among the files stored on said storage medium and following a procedure for erasing this file,

file compression means for selecting a file to be compressed from among the files stored on said storage medium and following a procedure for compressing this file, and

means for judging the necessity for erasure or compression of a file on said storage medium based on the free space on said storage

medium, and selectively executing either said file erasure means or said file compression means if file erasure or compression has been judged necessary.

13. A computer system for storing and managing a plurality of files using a storage medium,

characterized by providing file erasure means for selecting a file to be erased from among the files stored on said storage medium and following a procedure for erasing this file,

file compression means for selecting a file to be compressed from among the files stored on said storage medium and following a procedure for compressing this file,

means for judging the necessity for erasure or compression of a file on said storage medium based on the free space on said storage medium, and

means for giving said file compression means priority over said file erasure means until there are no more files to be compressed from said storage medium if these judgment means have judged file erasure or compression necessary.

14. A computer system for managing input and output of files to and from a storage medium capable of storing a plurality of files,

characterized by providing judgment means for judging the necessity for erasure or compression of a file on said storage medium based on the free space on said storage medium,

file erasure means for selecting a file to be erased from among the

files stored on said storage medium according to the priority of these files if these judgment means have judged file erasure or compression necessary,

file erasure means for following a procedure for erasing the file selected by these file erasure selection means,

file compression means for selecting a file to be compressed from among the files stored on said storage medium according to the priority of these files if these judgment means have judged file erasure or compression necessary,

file compression means for following a procedure for compressing the file selected by these file compression selection means, and

means for selectively executing said file erasure means or said file compression means based on the relationship between said selected file to be erased and said selected file to be compressed.

15. A file managing method for managing the free space on a storage medium of a computer system,

characterized by judging the necessity for erasure or compression of a file on said storage medium based on the free space on said storage medium,

selecting a file to be compressed from among the files stored on said storage medium and following a procedure for compressing this file if these judgment means have judged file erasure or compression necessary, and

selecting a file to be erased from among the files stored on said

storage medium and following a procedure for erasing this file if there are no files which may be compressed.

16. A file managing method for managing the free space on a storage medium of a computer system,

characterized by judging the necessity for erasure or compression of a file on said storage medium based on the free space on said storage medium,

selecting a file to be erased and a file to be compressed from among the files stored on said storage medium according to the priority of these files if these judgment means have judged file erasure or compression necessary, and

following a procedure for erasing said file to be erased or following a procedure for compressing said file to be compressed based on the relationship between said selected file to be erased and said selected file to be compressed. /4

DETAILED EXPLANATION OF THE INVENTION

INDUSTRIAL FIELD OF APPLICATION

This invention pertains to a computer system and a file managing method used by this system.

PRIOR ART

Data held indefinitely in a computer system are usually stored on a nonvolatile storage medium such as a disk by logical units called files. The software for allocating the logical data units called files to different recording sites on a physical nonvolatile

storage medium is called a file system.

Because data are stored using a physical storage medium, file systems have limited memory capacity. Previously, if a user wanted to create a new file, but there was no free space, the user had to explicitly indicate which file should be erased, and execute a file erasure program to create free space. This program indicated a file to the operating system and issued an erasure request, upon receiving which, the operating system nullified this file stored in the file system to create free space. That is, if there was no or little free space due to an increase in the size of the files stored in a file system, the user had to select which file should be erased and start up a program which explicitly issued an erasure request to the operating system to erase the file and create free space.

PROBLEMS THAT THE INVENTION IS TO SOLVE

As discussed above, previously, if there was not enough free space on a storage medium, the user has had to explicitly issue an erasure request to erase a file.

As a result, the user has had to carry out an operation to select which file should be erased or memorize how to use a program for erasing, which has been a source of poor computer user friendliness.

Another problem is that if a user was not aware that there was no free space when transmitted data arrived externally through a network or the like, the file system could not store the transmitted

data, resulting in losing reception of data.

This invention was developed upon reflecting on such problems. Its purpose is to provide a computer system and a file managing method which can automatically assure free space for storing a file on a storage medium and effectively use the recording capacity of a limited physical storage medium without the user issuing an erasure request or the like.

MEANS OF SOLVING THE PROBLEMS

This invention is a computer system for managing input and output of files to and from a storage medium capable of storing a plurality of files, characterized by providing judgment means for judging the necessity for erasure or compression of a file on this storage medium based on the free space on this storage medium, and automatic file erasure/compression means for automatically selecting a file to be erased or compressed from among the files stored on this storage medium and following a procedure for erasing or compressing this file if these judgment means have judged file erasure or compression necessary.

This computer system judges whether to automatically erase or compress a file according to the size of the free space on a storage medium. A free space assurance processing using automatic file erasure or compression may be judged necessary if, for example, the free space on a storage medium is less than a specific capacity or the free space on a storage medium is less than the file size to be

written. If a free space assurance processing using automatic file erasure or compression is judged necessary, the system automatically selects a file to be erased or compressed from among the files stored on the storage medium. The system also follows a procedure for erasing or compressing this selected file. This file erasure/compression procedure may actually erase or compress the file by file erasure/compression means, or issue an erasure or compression request to an operating system and have this operating system erase the file. This can automatically produce free space on a storage medium for storing files without the user explicitly issuing an erasure or compression request, and effectively use the limited recording capacity of a physical storage medium. Thus, this does not require that a user select which file should be erased or which file should be compressed or memorize how to use a program for erasing or compressing, and can realize a computer system with better user-friendliness. When transmitted data arrive from a network or another computer, this can also save the data without loss without the user creating free space.

The judgment whether to follow a free space assurance processing using automatic file erasure or compression is preferably processed each time that a user program or the like issues a file writing request or at regular intervals. This can assure the maximum free space at all times, safeguard new files which a user program has commanded be written from inadequate recording capacity, and prevent

the problem of loss of files downloaded from a network or the like. /5

The selection procedure of a file to be erased or compressed preferably selects a file to be erased or compressed according to a priority of files considering how much effect file erasure or compression will have on the user. This "priority of files" is an index indicating how much effect file erasure or compression will have on the user. The priority preferably uses priority information for each file indicated beforehand by the user or input/output history regarding, for example, whether a file has been read from the storage medium, the number of times read, and the last date and time read, allowing the user using this computer system to judge which files are important.

The user may indicate priority by the file name given to a file when saving a new file, where the "file name" given to a file has a separate dedicated parameter indicating priority. Such user-indicated priority information can be used to select a file to be erased based on the user's own judgment of importance. The file extension can also be used for priority, in which case, priority can be assigned according to the type of file, such as program file, system file, text file, compressed file, or graphics file, or the relationship to the program handling the file.

Setting the priority of all files higher than a reference value for selecting a file for erasure or compression and dynamically updating this reference value if there is no applicable file for

erasure or compression allows the file with the lowest priority of the files greater than this reference value to be selected as a file to be erased or compressed.

A user can usually judge which files are most important to the user by which files have been inputted or outputted most frequently or which files have been referred to most recently. Although files which the user has already referred to once may be erased in the case of data files such as news or weather reports downloaded from a network or the like, there are applications for which data not yet referred to must not be erased. Therefore, selecting a file to be erased based on the input/output history of a file, especially the file reference history on reading the file, can as much as possible avoid erasing files which if erased would have a great effect on the user; that is, files which may be expected to be read by the user in the future.

The priority may also be by file size, with the largest files erased first. This can assure the required free capacity while erasing the fewest number of files.

The computer system of this invention is also characterized by the capacity to store file correlation information indicating correlations between file pairs with interchangeable content among the files stored on this storage medium (for example, files with the same content differing only in file format, or a data file and its compressed data file), and referring to this file correlation

information to select a file which is recoverable from the content of another file as a file to be erased or compressed. This can efficiently erase files without any effect whatsoever on the user. Although especially ideal when erasing files, this configuration can also be used for compressing files.

For files correlating with other files such as hypertext files, files which are referred to more often by other files have a greater effect when erased or compressed. Therefore, a preferred mode detects the number of references by other files, such as the number of other files referring to this file, for each file stored on a storage medium, and selects a file to be erased or compressed from among the files stored on said storage medium according to this detected number of references.

The computer system of this invention can also use a suitable combination of means in addition to the means discussed above to realize an automatic file erasure method which has less effect on the user, such as notifying the user of files selected by the file erasure selection means and obtaining permission for erasure before erasing.

The computer system of this invention is also characterized by providing both an automatic file erasure means and an automatic file compression means, and using these means selectively. Compressing a file usually has less effect on a user than erasing a file, but erasing a file can secure greater free space more easily than

compressing a file. Therefore, files can be managed more efficiently by, for example, selectively compressing or erasing based on explicit commands by the user, or automatically selecting compression or erasure based on the size of the free space on a storage medium.

This invention is also characterized by providing both an automatic file erasure means and an automatic file compression means, giving automatic file compression priority until there are no more files to be compressed (for example, until all files have been compressed, or the only files still uncompressed are files which must never be compressed according to the directory, file extension, or the like), and turning to automatic file erasure when there are no more files to be compressed. Assigning use of compression and erasure in this way can lessen the effect on the user.

Selecting both files to be erased and files to be compressed according to separate criteria and switching between file erasure and file compression according to the relationship between the selected files can also assure the optimum free space. /6

WORKING EXAMPLES OF THE INVENTION

Next, working examples of this invention will be discussed referring to the appended drawings.

Fig. 1 shows the configuration of a file system used in a computer system according to a first working example of this invention. This file system (2) has a function for realizing either one of automatic file erasure or automatic file compression as a

function for automatically assuring free capacity on a recording medium. This file system (2) is realized as part of an operating system. A computer system applying this file system (2) comprises a conventional computer having hardware resources such as a CPU, a memory, an auxiliary storage device, and various I/O devices. The automatic file erasure/compression function of the file system (2) is ideal, however, for electronic information devices such as portable computers in which the auxiliary storage device has relatively small capacity for file storage. Therefore, the computer system will be discussed below by examples applied to portable electronic information devices such as PDA or subnotebook computers.

As shown in Fig. 1, the electronic information device of this first working example comprises user programs (1) such as application programs to be executed by the CPU of this device, a file system (2) for receiving requests by these programs (1) for file operations and managing input and output of files, and a nonvolatile storage medium (3) such as a hard disk or a flash memory card used as an auxiliary storage medium for storing files.

File operations may be requested not only by the user programs (1), but also by commands issued on the OS level in response to key input operations by the user. In either case, however, all requests come from other programs as viewed by the file system (2). Therefore, a request by the user will be handled here as a request by a program.

The file system (2) is provided with a request reception part (21), an erasure/compression conditions judgment part (22), a file erasure/compression selection part (23), a file erasure part (24), and a file input/output part (25) as shown in Fig. 1 to realize an automatic free space assurance function using file erasure or file compression.

The request reception part (21) receives requests by the user programs (1) for standard file operations such as reading or writing a file. The erasure/compression conditions judgment part (22) is for judging based on the free space on the storage medium (3) whether a file on the storage medium (3) must be erased or compressed to assure free space; that is, whether the conditions for free space assurance processing using file erasure or compression (erasure/compression conditions) have been met. This part processes judgment as to whether erasure/compression conditions have been met each time a file operation request such as file writing has been issued to the request reception part (21), or regularly at specific time intervals.

The file erasure/compression selection part (23) is started whenever a judgment is made that erasure/compression conditions have been met, and selects a file to be erased or compressed from among the files (including data files and program files) stored on the storage medium (3) based on, for example, the priority of these files. "Priority" here is an index indicating how much effect processing erasure or compression of this file will have on the user,

where the greater the effect of the file, the higher the priority.

The file erasure/compression part (24) processes a selected file for either erasure or compression. This part processes a file selected to be erased for erasure from the storage medium (3) if it has an automatic erasure function, and processes a file selected to be compressed for compression if it has an automatic compression function. "File erasure" and "file compression" are equivalent here. The term "file deletion" may sometimes be used instead of "file erasure" depending on the operating system.

Although a file to be erased may be physically erased (deleted) from the storage medium (3) when erasing a file, means such as moving to a dedicated storage area called a "trash bin" may also be used, as when erasing files in conventional file systems, to just nullify the management data for this file, such as the directory entry, while leaving the physical file untouched. Needless to say, an external storage device can be connected to a portable computer and used for backup, in which case, this file to be erased may be erased from the storage medium (3) and moved to the external storage device as a backup file.

Fig. 2 shows the configuration of files managed by this file system (2).

Files are managed by a tree hierarchy in this working example. Specifically, several child nodes are held beneath a given node, and the parent node is uniformly fixed as viewed from each node. A node

with no child nodes is a file, which holds data. A node with child nodes is called a directory. The directory at the highest level of the tree hierarchy is called a route, and is its own parent node.

Fig. 2 shows four files with the file names of "616.1" to "616.4" as child nodes of a "news" directory, and three files with the files names of "95.1" to "95.3" as child nodes of a "schedule" directory. /7

Fig. 3 shows an example of an actual file management hierarchy corresponding to Fig. 2.

The first field of each node in Fig. 3 indicates the type of the node. D is a directory, and F is a file. The second field is the name of the file or the directory. In the case of a directory, the third and subsequent fields are pointers to child nodes. Diagonal lines in the drawing indicate the end, which usually records a value not used as a pointer. For example, if the number of a node is used as a pointer, minus 1 may be used to record the end.

In the case a file, the third field stores a value indicating the importance of this file. This importance value is used as the priority discussed above, and is referred to when selecting a file to be erased or a file to be compressed. Importance is expressed in Fig. 3 by values from 1 to 10, with higher values indicating greater importance. This importance value is indicated by the user when creating or saving this file. The fourth field indicates the actual data size, and the fifth field is a pointer to the data.

Fig. 4 shows an algorithm giving a procedure for file management processing by the file system (2).

First, the request reception part (21) of the file system (2) receives a file operation request from a user program (1) (step S11). If this file operation request is not a file writing request (step S12), the file operation indicated by this file operation request (such as file reading, file erasure, or directory reference) is carried out as usual (step S13).

If the file operation request from the user program (1) is a file writing request (step S12), however, the erasure/compression conditions judgment part (22) judges whether a specific erasure/compression condition has been met based on the free capacity of the storage medium (3) (step S14). In this example, the judgment part compares the file size of the writing request to the current free capacity of the storage medium (3), and judges that the erasure/compression condition has been met when the size of the free space is less than the file size of the writing request. The judgment part may also judge that an erasure/compression condition has been met if the current free capacity of the storage medium (3) is less than a specific capacity regardless of the file size of the writing request.

The current free capacity of the storage medium (3) is stored in the file system (2). Specifically, the file system (2) decreases the total storage capacity of the storage medium (3) allocated to this

file system (2) by exactly the file size each time that a file is written, and stores this result as the current free capacity of the storage medium (3).

If there is enough free capacity and the erasure/compression condition has not been met (step S15), the requested file is immediately written (step S16). If there is not enough free capacity and the erasure/compression condition has been met (step S15), however, the file erasure/compression selection part (23) processes file selection (step S17).

Specifically, the file erasure/compression selection part (23) selects a file with a lower importance than a reference value according to the priority (importance in Fig. 3) given to each of the files stored in the storage medium (3). The file erasure/compression part (24) then erases or compresses the selected file (step S18).

Next, the erasure/compression conditions judgment part (22) again judges whether the erasure/compression condition has been met (step S14), and repeats the processing of the steps S17 and S18 until the condition is no longer met. File writing is processed in step S16 at the point when the erasure/compression condition is no longer met.

Fig. 5 shows an example of a user program (1) when making a file writing request to the request reception part (21).

First, the program (1) requests creation of a new file (file create). In this case, the program indicates a new file name ("/schedule/95.6") and sends a system call for creating a new file to

the file system (2) as shown in Fig. 5(1). Upon successfully creating a file, the file system (2) returns a file identifier (fd) to the program (1). Next, the program (1) requests writing to this file. The programs asks the file system (2) to write 256 bytes for the file fd indicated by the file identifier starting from the position in the program specified by the variable buf. That is, the writing request indicates the file to be written and the number of bytes to write.

Next, the erasure/compression conditions judgment part (22) judges whether a predetermined erasure/compression condition has been met. In this case, as discussed above, the judgment part judges that an erasure/compression has been met if the free space on the storage medium (3) is less than the file size of the writing request or the free space is less than a predetermined size.

Fig. 6 shows a specific example of a second judgment system by the erasure/compression conditions judgment part (22).

In this judgment processing, if the file system (2) receives a writing request when the utilization rate of the storage medium (3) is greater than a specific rate RU, the file system repeats free space assurance processing using file erasure or compression until the utilization rate is less than another specific rate SL. The /8 utilization rate of the storage medium (3) increases when files are written, and decreases when files are erased, partially deleted, or compressed. Fig. 6 shows a situation when files were erased or decreased when a writing request was received at the times t1 and t2

because the utilization rate was greater than SU.

Fig. 7 shows an algorithm corresponding to the erasure judgment processing in Fig. 6.

The erasure/compression conditions judgment part (22) calculates the current utilization rate SR of the storage medium (3) (step S21) and investigates whether this rate is at least SU (step S22). If SR ≥ SU, the file erasure/compression selection part (23) is activated to select a file to be erased or compressed (step S23). Next, after the file erasure/compression part (24) has erased or compressed this selected file, the erasure/compression conditions judgment part (22) calculates the utilization rate SR again (step S24). The erasure/compression conditions judgment part (22) then investigates whether SR ≥ SU, and repeats the steps S23 and S24 if SR is less than SL. Usually, SL < SU.

If the erasure/compression conditions judgment part (22) judges that the erasure/compression condition has not been met, the writing request is processed as usual. That is, writing is carried out if there is no problem after checking a writing limit or the like.

If the erasure/compression conditions judgment part (22) judges that the erasure/compression condition has been met, the file erasure/compression selection part (23) selects a file to be erased or compressed. Several algorithms may be considered for this selection.

For example, the selection part may select one file with the lowest importance from the files found in the file system. If there are several files with the lowest importance, the selection part may select any one file, the file with the largest file size of these, or the file with the smallest file size. Alternately, the selection part may select the file with the smallest file size within a range no greater than the file size of the writing request.

Specific files which should not be erased or compressed may be handled by setting the file erasure/compression selection part (23) so as not to select a file with importance greater than a certain level for erasure or compression, and giving files which should not be erased or compressed an importance of greater than this set level. For example, if files greater than importance level 9 are not to be erased or compressed, "95.1," "95.2," and "95.3" of the files shown in Fig. 3 are not to be erased or compressed. This can remove specific files from the files to be erased or compressed. This importance level need not be fixed, but can be variable.

Even if the erasure/compression conditions judgment part (22) has judged the necessity of erasure or compression, the file erasure/compression selection part (23) may be unable to select a file for erasure or compression. In this case, a handling such as the following is desirable.

(1) The file erasure/compression selection part (23) replies to the program (1) or notifies the user that there are no applicable

files, and does not erase or compress any files.

(2) If the system of not selecting files of greater than a predetermined level of importance has been adopted, the file erasure/compression selection part (23) changes this set level to a lower level if it cannot select a file. Alternately, the selection part selects the file with the lowest importance from among the files of greater than the predetermined level of importance. In these cases, the selection part preferably notifies the user and obtains permission before selecting a file.

Fig. 8 shows the system configuration of an electronic information device according to a second working example of this invention.

This file system has an additional erasure/compression notification part (26). This part is configured so as to notify the user of the file selected to be erased or compressed.

Specifically, the erasure/compression notification part (26) sends the file selected by the file erasure/compression selection part (23) to an erasure/compression notification program (4) for notification to the user. This allows the user to know that a file will be automatically erased or compressed, and which file will be erased or compressed. The timing of the notification by the erasure/compression notification part (26) may be either before or after the file erasure/compression part (24) erases or compresses the file.

Fig. 9 shows the system configuration of an electronic information device according to a third working example of this invention.

The file system (2) is provided with a permission information reception part (27) in addition to the configuration of the second working example.

Specifically, the erasure/compression notification part (26) sends the file selected by the file erasure/compression selection part (23) to the erasure/compression notification program (4) for notification to the user. The user inputs to the erasure/compression notification program (4) whether erasure or compression is permitted or not permitted. The permission information reception part (27) receives and inputs this permission information to the file erasure/compression part (24). If the user permits free space assurance processing by erasing or compressing this file, this file is erased or compressed. If not, the file erasure/compression selection part (23) selects another file. As a result, a file is not erased or compressed without the user's knowledge even if labeled as a file of lower importance, but is erased or compressed only after receiving permission from the user. Fig. 10 shows a flowchart corresponding to Fig. 9.

First, the request reception part (21) of the file system (2) receives a file operation request from a user program (1) (step S31). If this file operation request is not a file writing request (step

S32), the file operation indicated by this file operation request /9
(such as file reading, file erasure, or directory reference) is carried out as usual (step S33).

If the file operation request from the user program (1) is a file writing request (step S32), however, the erasure/compression conditions judgment part (22) judges whether a specific erasure/compression condition has been met based on the free capacity of the storage medium (3) (step S34). In this example, the judgment part compares the file size of the writing request to the current free capacity of the storage medium (3) as discussed above, and judges that the erasure/compression condition has been met when the size of the free space is less than the file size of the writing request. The judgment part may also judge that an erasure/compression condition has been met if the current free capacity of the storage medium (3) is less than a specific capacity regardless of the file size of the writing request. The utilization rate SR discussed above can be used as the free space.

If there is enough free capacity and the erasure/compression condition has not been met (step S35), the requested file is immediately written (step S36). If there is not enough free capacity and the erasure/compression condition has been met (step S15), however, the file erasure/compression selection part (23) processes file selection (step S37). Specifically, the file erasure/compression selection part (23) selects a file with a lower importance than a

reference value according to the priority (importance in Fig. 3) given to each of the files stored in the storage medium (3).

Next, the file erasure/compression part (24) sends this selected file to the erasure/compression notification program (4) for notification to the user (steps S38 and S39). The user inputs to the erasure/compression notification program (4) whether erasure or compression is permitted or not permitted. The permission information reception part (27) receives (step S40) and inputs this permission information to the file erasure/compression part (24). If the user permits free space assurance processing by erasing or compressing this file (step S41), the file erasure/compression part (24) erases or compresses this file. If not, the steps S37 to S41 are repeated, and the file erasure/compression selection part (23) selects another file.

Fig. 11 shows another example of this file priority management.

Specifically, each file in Fig. 3 was provided with an importance field where the importance of the file was stored, but in Fig. 11, a table pairing file names with importance is prepared, which may be stored in a separate file. This table is updated each time a file is created, deleted, or changed in importance.

Priority may also be managed by the file name given by the user to a new file when this file is created and saved. The file extension can also be used for priority, in which case, priority can be assigned according to the type of file, such as program file, system

file, text file, compressed file, or graphics file, or the relationship to the program handling the file.

First to third working examples were discussed above in which a file system (2) was incorporated as a file managing part in an operating system, but the file erasure/compression function of a file system (2) can also be realized as a program running on the user level outside the operating system. Fig. 12 shows an example of the system configuration of a portable computer in this case.

Specifically, in Fig. 12, the file erasure/compression function of the file system (2) discussed in the first to third working examples is housed as a file management program (6) running outside the operating system. This program runs separately from the conventional user programs (1), and does not receive requests directly from the conventional user programs (1). As viewed by the operating system (5), the file management program (6) can be seen as a user program just like the conventional programs (1).

Fig. 13 shows the functional configuration of a file management program (6).

The erasure/compression conditions judgment part (31), file erasure/compression selection part (32), and file erasure/compression part (33) in Fig. 13 correspond to the erasure/compression conditions judgment part (22), file erasure/compression selection part (23), and file erasure/compression part in Fig. 1, but the input/output part in Fig. 13 functions as an interfaced with the operating system (5).

Upon judging that the erasure/compression condition has been met, the erasure/compression conditions judgment part (31) activates the file erasure/compression selection part (32), which selects a file to be erased or compressed. The file erasure/compression part (33) issues either an erasure request or a compression request to erase or compress the selected file. This request is sent through the input/output part (34) to the operating system (5) as a system call. The operating system (5) receives this request and erases or compresses the file. The specific processing followed by the erasure/compression conditions judgment part (31) and the file erasure/compression selection part (32) is the same as the processing discussed in Figs. 6 and 7.

As discussed above, by providing either an automatic file erasure function or an automatic file compression function, a file system (2) (including a file management program) used by an electronic information device of the first to third working examples can effectively use the limited storage capacity of a physical storage medium (3) so as to automatically create a free space on the storage medium (3) storing files without the user explicitly issuing an erasure request or a compression request. Therefore, this can realize a computer system with good user friendliness without requiring the user to select which files should be erased or memorize how to use a program for erasure when there is little free space. This file system can also store data transmitted by a network or

/10

another computer without loss of transmitted data and without the user creating a free space.

In this discussion, whether the erasure/compression condition has been met was judged when a user program (1) issued a file writing request, but this judgment may also be processed at a specific time interval. The configuration of Fig. 13, which is independent of the user programs (1), is ideal for this.

Next, a fourth working example of this invention will be discussed referring to Figs. 14 to 21.

In the first to third working examples discussed above, a fixed predetermined importance for each file was used as a priority for file erasure/compression selection, but the fourth working example is configured to use the input/output history of each file, that is, the use status of a file by the user, as a priority for file erasure/compression selection. Such a priority is used for the following reasons:

A computer system usually has files which are frequently referenced and files which are not frequently referenced. Erasing a frequently referenced file has a high probability of causing the problem that the user will not find the file when the user wishes to refer to this file. Similarly, compressing a frequently referenced file causes the problem of requiring expansion processing each time this file is read, lengthening the time required to read the file.

Conversely, text files such as news downloaded from a network or

the like can be expected to be files which have low importance to the user once they have been read. Hence, this is an application in which files already referred to once by the user may be erased, but files not yet referred to should not be erased.

Therefore, to handle such situations, the fourth working example uses file reading history information for priority. Next, the configuration of the fourth working example will be discussed.

Like the system configuration of the first to third working examples, an electronic information device of this fourth working example comprises user programs (10) such as application programs to be executed by the CPU of this device, a file system (20) for receiving requests by these programs (10) for file operations and managing input and output of files, and a nonvolatile storage medium (30) such as a hard disk or a flash memory card used as an auxiliary storage medium for storing files.

The file system (20) comprises a file input/output part (201), a file reading part (202), a file reading history creation part (203), a file erasure/compression selection part (204), and a file erasure/compression part (205).

When reading a file, besides reading the file from a file storage part (31) of the storage medium (30) through the file input/output part (201), the file reading part (202) notifies the file name of this file and the fact that this file has been read to the file reading history creation part (203). The file reading

history creation part (203) creates file reading history information, indicating, for example, whether the file has been read, how many times it has been read, and the last reading date and time, for each file based on the notification from the file reading part (202). This file reading history information is stored through the file input/output part (201) in a file reading history storage part (32) of the storage medium (30).

The file erasure/compression selection part (204) refers to the file history reading information in the file reading history storage part (32) through the file input/output part (201), selects a file which is expected to be unimportant to the user to be erased or compressed from among the files stored in the file storage part (31), and notifies the name of this file to the file erasure/compression part (205). As in the first to third working examples, this file erasure/compression selection part (204) may process file erasure/compression selection each time that a program (10) issues a file operation request such as a file writing request, or at regular intervals.

The file erasure/compression part (205) erases or compresses the selected file. This file erasure or compression is done by the same system as in the first to third working examples.

Next, file reading history information will be discussed in detail referring to Figs. 15 to 17.

In the example in Fig. 15, the file reading history information is stored as one file management information in a directory. The file reading history information is expressed as a flag of "0" or "1", where "0" indicates a file which has not been read, and "1" indicates a file which has been read.

In the example in Fig. 16, how many times a file has been read is used as file reading history information instead of a flag indicating whether a file has been read. This is stored in a directory of file correlations in the same way as in Fig. 15.

The file reading history information in these examples in Figs. 15 and 16 can be stored as a reading history file separate from the directory as shown in Fig. 17. The last reading date and time of the file can be managed in the same way as whether and how many times the file has been read. The operating system also has a function for managing the last reading date and time of each file, which may also be used. /11

Next, the automatic file erasure/compression operation using file reading history will be discussed referring to Fig. 18.

First, the same procedure as in the first to third working examples is used to investigate whether the file storage part (31) of the storage medium (30) has enough free space; that is, whether the erasure/compression condition has been met (step S51). If the erasure/compression condition has been met, the processing of the file erasure/compression selection part (204) is started.

Specifically, the file erasure/compression selection part (204) first fetches the reading history information for one file from the file reading history storage part (32) (step S53), and investigates whether this file has not been read or the number of times read is no more than a specific value (step S54). If this file has not been read or the number of times read is no more than the specific value, the file erasure/compression selection part (204) transmits the name of this file to the file erasure/compression part (205) to be erased or compressed (step S55).

Next, whether the erasure/compression condition has been met is investigated again (step S51), and the steps S53 to S55 are repeated until the erasure/compression condition is no longer met or the reading history information for all files has been read.

Using reading history information to determine a file to be erased or compressed in this way can as much as possible prevent erasing or compressing files which have been read frequently and may be expected to be read in the future. Therefore, this can greatly reduce the effect of erasure/compression on the user.

Fig. 19 shows another example of file reading history information.

This example uses the last reading date and time information for each file for the file reading history. When reading a file in this example, besides reading the file from the file storage part (31), the file reading part (202) notifies the name of this file to the

file reading history creation part (203). The file reading history creation part collates the notified file name with the current date and time, and stores this in the file reading history storage part (32).

Next, the automatic file erasure/compression operation using file reading history will be discussed referring to Fig. 20. Basically, files are erased or compressed by prioritizing in the order of the files with the oldest last reading date and time.

First, the same procedure as in the first to third working examples is used to investigate whether the file storage part (31) of the storage medium (30) has enough free space; that is, whether the erasure/compression condition has been met (step S61). If the erasure/compression condition has been met, the processing of the file erasure/compression selection part (204) is started. Specifically, the file erasure/compression selection part (204) first fetches the reading history information for one file from the file reading history storage part (32) (step S63), and investigates whether the last reading date and time is older than a specific predetermined date and time (step S64). If the last reading date and time is older than the specific predetermined date and time, the file erasure/compression selection part (204) transmits the name of this file to the file erasure/compression part (205) to be erased or compressed (step S65).

Next, whether the erasure/compression condition has been met is investigated again (step S61), and the steps S63 to S65 are repeated until the erasure/compression condition is no longer met or the reading history information for all files has been read.

Recording the last reading date and time in this way can as much as possible prevent erasing or compressing files which have been referenced recently, which can efficiently assure free space. Combining the last reading date and time with the number of times read discussed above can remove files with recent reading dates and times from the erasure/compression object even if these have been read only a few times, which can assure free space even more efficiently.

Next, another example of an automatic file erasure/compression operation using the reading history information of Fig. 15 (whether a file has been read) will be discussed referring to Fig. 21. This example preferentially erases or compresses files which have been read.

First, the same procedure as in the first to third working examples is used to investigate whether the file storage part (31) of the storage medium (30) has enough free space; that is, whether the erasure/compression condition has been met (step S71). If the erasure/compression condition has been met, the processing of the file erasure/compression selection part (204) is started. Specifically, the file erasure/compression selection part (204) first

fetches the reading history information for one file from the file reading history storage part (32) (step S73), and investigates whether this is a file which has already been read (step S74). If this file has already been read, the file erasure/compression selection part (204) transmits the name of this file to the file erasure/compression part (205) to be erased or compressed (step S75).

Next, whether the erasure/compression condition has been met is investigated again (step S71), and the steps S73 to S75 are repeated until the erasure/compression condition is no longer met or the reading history information for all files has been read. /12

An algorithm which preferentially erases or compresses files which the user has already referred to once in this way is especially effective and can efficiently use the storage medium of an electronic information device with limited storage capacity in the application of downloading news or weather reports or the like through a telephone line, a network, or a medium such as radio waves so that the user can refer to these downloaded files as needed.

Fig. 22 shows the system configuration of an electronic information device according to a fifth working example of this invention.

This system adds a function for erasing the corresponding information in the file reading history storage part (32) when writing a file to what is roughly the file system (20) of the fourth working example. Specifically, the file system (20) used in this

fifth working example is the same as in the fourth working example in terms of referring to file reading history information to select a file to be erased or compressed, but has a file writing part (206) added to increase the reliability of this file reading history information. Fig. 23 shows the operation for erasing file reading history information using this file writing part (206).

As shown in the flowchart in Fig. 23, when writing a file, first, the file writing part (206) writes the file in the file storage part (31), and notifies the name of this file to the file reading history creation part (203) (steps S81 and S82). The file reading history creation part (203) erases or resets to its initial settings the file reading history information of the file indicated by this file name (step S83).

For example, if using file reading history information indicating whether a file has been read as shown in Fig. 24, when writing to the file with the file name f3, the file reading history information for this file name f3 is reset to "0", which treats this file as a file which has not been read.

Initializing the corresponding file reading history information when writing a file in this way can consider whether the user has read data after updating when selecting a file to be erased in an application in which data which the user has already referred to once may be erased, but data not yet referred to must not be erased.

Fig. 25 shows the system configuration of an electronic information device according to a sixth working example of this invention.

The file system (20) used in this system is provided with a file input/output part (301), a file writing part (302), a file size information creation part (303), a file erasure/compression selection part (304), and a file erasure/compression part (305) as shown in the drawing to realize an automatic free space assurance function using file erasure or file compression.

When writing a file, besides writing a file to the file storage part (31) through the file input/output part (301), the file writing part (302) notifies the name of this file and the file size to the file size information creation part (303). The file size information creation part (303) creates a table pairing the file name with the file size, and records this table in a file size information storage part (33) through the file input/output part (301). This file size information can be stored in part of the directory entry for each file. The file erasure/compression selection part (304) refers to the file size information storage part (33) to obtain the size of each file when erasing or compressing a file. The selection part then selects files of greater size than a certain set size, and transmits these file names to the file erasure/compression part (305). The file erasure/compression part (305) erases the files with the file names obtained from the file erasure/compression selection part (304) from

the file storage part (31).

Next, the automatic file erasure/compression operation using file size information will be discussed referring to Fig. 26. In this example, files with the largest size are preferentially erased or compressed.

First, the same procedure as in the first to third working examples is used to investigate whether the file storage part (31) of the storage medium (30) has enough free space; that is, whether the erasure/compression condition has been met (step S91). If the erasure/compression condition has been met, the processing of the file erasure/compression selection part (304) is started. Specifically, the file erasure/compression selection part (304) first fetches the size information for one file from the file size information storage part (33) (step S93) and investigates whether the size of this file is greater than a predetermined reference size (step S94). If the file size is greater than the reference size, the file erasure/compression selection part (304) transmits the name of this file to the file erasure/compression part (305) to be erased or compressed (step S95).

Next, whether the erasure/compression condition has been met is investigated again (step S91), and the steps S93 to S95 are repeated until the erasure/compression condition is no longer met or the size information has been read for all files.

Preferentially erasing or compressing the files with the largest size in this way can minimize the number of files to be erased or compressed to free up the capacity of the file storage part (31), and /13 reduce the probability of causing the problem that a file which the reader is about to read has already been erased, or compressed and will take time to read.

The processing for preferentially erasing or compressing the files with the largest size in this way can also be combined with the file reading history information discussed in the fourth and fifth working examples. A preferred method in this case is to first refer to the file reading history information to select several candidate files, then erase or compress the files with the largest size from among these candidates.

Fig. 27 shows the system configuration of an electronic information device according to a seventh working example of this invention.

The file system (20) used in this system is provided with a file input/output part (401), a file conversion part (402), a file correlation creation part (403), a file erasure/compression selection part (404), and a file erasure/compression part (405) as shown in the drawing to realize an automatic free space assurance function using file erasure or file compression.

The file conversion part (402) reads the content of a certain file fa from the file storage part (31), subjects the content to

conversion (such as compressing or expanding file data, or converting to a text file mode), then stores the converted file in the file storage part (31) as a separate file fb. The file conversion part (402) then notifies the file correlation creation part (403) that the files fa and fb have interchangeable content and differ only in data mode.

The file correlation creation part (403) creates file correlation information indicating the correlation between two files with interchangeable content stored in the file storage part (31) based on the notification from the file conversion part (402), and stores this information in the file correlation storage part (34).

The file extension registered in the directory of each file can be used as file correlation information. Specifically, for example, giving the same file name as a certain file fa to a compressed file fb obtained by compressing this file fa and using information indicating the data compression mode as an extension indicating the file type allows one to recognize that the files fa and fb have interchangeable content from the relationship between the file names and the extensions of the files fa and fb. When employing this system, the file correlations storage part (34) can be realized as part of the directory.

When erasing or compressing a file, the file erasure/compression selections part (404) first refers to file correlation storage part (34) to select a file which can be restored from the content of

another file, and transmits the name of this file to the file erasure/compression part (405). The file erasure/compression part (405) erases or compresses the file with the file name obtained from the file erasure/compression selection part (404) from the file storage part (31).

Next, the automatic file erasure/compression operation using file correlation storage information will be discussed referring to Fig. 28. In this example, files which can be restored by converting from other files are preferentially erased or compressed.

First, the same procedure as in the first to third working examples is used to investigate whether the file storage part (31) of the storage medium (30) has enough free space; that is, whether the erasure/compression condition has been met (step S101). If the erasure/compression condition has been met, the processing of the file erasure/compression selection part (404) is started. Specifically, the file erasure/compression selection part (404) first fetches the information for one file from the correlation information storage part (34) (step S103) and investigates whether this file can be restored from the content of another file (step S104). If this file is restorable, the file erasure/compression selections part (404) transmits the name of this file to the file erasure/compression part (405) to be erased or compressed (step S105).

Next, whether the erasure/compression condition has been met is investigated again (step S101), and the steps S103 to S105 are

repeated until the erasure/compression condition is no longer met or the size information has been read for all files.

Managing in this way so as to preferentially erase or compress files the contents of which can be restored by converting from other files and not processing erasure or compression for files which are not restorable is especially effective when employing automatic file erasure for assuring free space, and can reduce the incidence of the problem that a file which the user is about to read has already been erased and cannot be read. Processing to erase or compress restorable files in this way can also be combined with the file reading history information discussed in the fourth and fifth working examples. A preferred method in this case is to first refer to the file reading history information to select several candidate files, then erase or compress restorable files from among these candidates.

Fig. 29 shows the system configuration of an electronic information device according to a eighth working example of this invention.

The file system (20) used in this system is provided with a file input/output part (501), a reference correlation reading part (502), a file erasure/compression selection part (503), and a file erasure/compression part (504) as shown in the drawing to realize an automatic free space assurance function which is ideal for files /14 which have reference correlations with other files, such as hypertext files.

When erasing or compressing a file, first, a command from the file erasure/compression selection part (503) starts the operation of the reference correlation reading part (502), which fetches the number of references by other files to each file in the file storage part (31). The number of references is detected by analyzing the content of the other files to investigate the number of pointers to the file in question. The file erasure/compression selection part (503) selects a file which has a lower number of references by other files than a specific number, and transmits the name of this file to the file erasure/compression part (503) to be erased or compressed from the file storage part (31).

Next, the automatic file erasure/compression operation using number of file references will be discussed referring to Fig. 30. In this example, files with the fewest number of references to this file by other files are preferentially erased or compressed.

First, the same procedure as in the first to third working examples is used to investigate whether the file storage part (31) of the storage medium (30) has enough free space; that is, whether the erasure/compression condition has been met (step S111). If the erasure/compression condition has been met, the processing of the file erasure/compression selection part (503) is started. Specifically, the file erasure/compression selection part (503) first fetches the information for one file from the file storage part (31) (step S113), and investigates the number of references to this file

by other files in the reference correlating reading part (502). The reference correlation reading part (502) detects the number of references by analyzing the contents of the other files to investigate the number of pointers to this file, and delivers this result to the file erasure/compression selection part (503). The file erasure/compression selection part (503) investigates whether the number of references for this file is fewer than a predetermined reference number (step S114). If the number of references is fewer than the reference number, the file erasure/compression selection part (503) transmits the name of this file to the file erasure/compression part (503) to be erased or compressed from the file storage part (31).

Next, whether the erasure/compression condition has been met is investigated again (step S111), and the steps S113 to S115 are repeated until the erasure/compression condition is no longer met or the size information has been read for all files.

Preferentially erasing or compressing files with the fewest references by other files and preserving the files with many references in this way can reduce the incidence of the problem that a file which the user wishes to read through references by other files has already been erased, especially when employing file erasure as a function for assuring free space. Processing to preferentially erase or compress files with a low number of references in this way can also be combined with the file reading history information discussed

in the fourth and fifth working examples. A preferred method in this case is to first refer to the file reading history information to select several candidate files, then preferentially erase or compress files with the fewest references by other files from among these candidates.

The automatic file erasure/compression function of the file systems (20) in the fourth through eighth working examples can be realized as a program running on the user level outside the operating system in the same way as the file systems (2) in the first through third working examples. In this case, only the file input/output parts in Figs. 14, 22, 25, 27, and 29 are installed in the operating system as file managing parts, and the other parts are installed as a file management program running outside the operating system.

As discussed above, electronic information devices according to the fourth to eighth working examples can automatically erase files with less effect on the user than using just a static priority by using, for example, the file reference history, size, correlations, or number of references to select files to be erased or compressed.

Electronic information devices according to the fourth to eighth working examples preferably use this priority suitably combined with a function for notifying the user of a file selected by the file erasure/compression selection part and obtaining erasure/compression permission before erasing or compressing this file as discussed in the second and third working examples.

Next, a ninth working example of this invention will be discussed.

Like the system configuration of the first to third working examples, an electronic information device of this ninth working example comprises user programs (100) such as application programs to be executed by the CPU of this device, a file system (200) for receiving requests by these programs (100) for file operations and managing input and output of files, and a nonvolatile storage medium (300) such as a hard disk or a flash memory card used as an auxiliary storage medium for storing files.

The file system (200) has both an automatic file erasure function and an automatic file compression function as functions for automatically assuring free capacity on the storage medium (300), and is configured so as to use this automatic erasure function or this automatic compression function selectively.

The file system (200) comprises a request reception part (601) for receiving requests from the programs (100), a file erasure selection part (602) for selecting files to be erased, a file erasure part (603) for erasing files, a file compression selection part (604) /15 for selecting files to be compressed, a file compression part (605) for compressing files, a file input/output part (606) for inputting or outputting files stored on the storage medium (300), and a control part (607) for governing these parts.

Fig. 32 shows the file structure managed by the file system (200).

The first field of each node indicates the type of the node. D is a directory, and F is a file. The second field is the name of the file or the directory. In the case of a directory, the third and subsequent fields are pointers to child nodes. Diagonal lines in the drawing indicate the end, which may use a value not used as a pointer. For example, if the number of a node is used as a pointer, minus 1 may be used to record the end. In the case a file, the third field indicates the last update date and time, and the fourth filed indicates the last reference date and time. The fifth field is a pointer to data.

Fig. 33 is a flowchart showing an algorithm for a file management system. In this working example, the last reference date and time is used as an attribute for selecting a file to be erased or compressed.

First, the request reception part (601) of the file system (200) receives a file operation request from a user program (100) (step S201). If this file operation request is not a file writing request (step S202), the file operation indicated by this file operation request (such as file reading, file erasure, or directory reference) is carried out as usual (step S203).

If the file operation request from the user program (100) is a file writing request (step S202), however, the control part (607)

judges whether a specific erasure/compression condition has been met based on the free capacity of the storage medium (300) (step S204). In this example, the control part compares the file size of the writing request to the current free capacity of the storage medium (300), and judges that the erasure/compression condition has been met when the size of the free space is less than the file size of the writing request. The control part may also judge that an erasure/compression condition has been met if the current free capacity of the storage medium (300) is less than a specific capacity regardless of the file size of the writing request.

If there is enough free capacity and the erasure/compression condition has not been met (step S205), the requested file is immediately written (step S206). If there is not enough free capacity and the erasure/compression condition has been met (step S205), however, the control part judges whether there is a compressible file on the storage medium (300) which may be compressed (step S207). This may be judged, for example, by investigating whether all files have been compressed. In this case, the control part judges that there is a compressible file if a remaining uncompressed file, and there is no compressible file if there is no remaining uncompressed file. Files which must not be compressed and files which may be compressed may also be determined explicitly by the user, for example, and set by means such as the directory or the file extension, and whether there is a compressible file may be judged based on this determination.

If there are remaining compressible files on the storage medium (300) which may be compressed, the file compression selection means (604) selects the file with the oldest last reference date and time (interpreting a file which has not been referenced as having the oldest last reference date and time) from among the files which may be compressed as a file to be compressed, and the file compression part (605) compresses this file (steps S208 and S209). Next, the control part judges again whether the file erasure/compression condition has been met (steps S204 and S205), and repeats the steps S208 and S209 so long as there are remaining files which may be compressed until the condition is no longer met.

If there are no compressible files, the processing for assuring free space then switches from file compression to file erasure.

In this case, the file erasure selection part (602) selects the file with the oldest last reference date and time from among all files as a file to be erased, and the file erasure part (603) erases this file (steps S210 and S211).

Next, the control part judges again whether the file erasure/compression condition has been met (steps S204 and S205), and repeats the steps S210 and S211 until the condition is no longer met.

Thus, this working example exercises control so as to give automatic file compression processing priority over automatic file erasure processing. As a result, the automatic free space assurance function can minimize the effect on the user.

A file which has not been referenced may be interpreted to have the oldest last reference date and time. Repeating this procedure can create free space at least equivalent to the file to be written.

Selecting a file to be compressed or erased in this example is not limited to using the same attribute of the latest reference date and time for both. Different attributes may be used by selecting the file with the oldest last reference date and time to be compressed, and the file with the largest file size to be erased.

Fig. 34 is a flowchart showing another algorithm for a file management system. By compressing all uncompressed files before erasing, for example, the example shown in Fig. 33 risks compressing a file with a high probability of being used next. Since file /16
compression consumes CPU time and electricity, and quickly returns to a state with no free space, it is poor policy to compress a file which is about to be used. To remedy this, it is better to not compress a file with a high probability of being used soon, leaving this uncompressed, and instead erase a file with a low probability of being used. Because it usually is difficult to judge whether a file will be used, files with a later last reference date and time are often judged to have a higher probability of being used on the theory of nearness in time. This assumption may be used in the present algorithm.

First, the request reception part (601) of the file system (200) receives a file operation request from a user program (100) (step

S301). If this file operation request is not a file writing request (step S302), the file operation indicated by this file operation request (such as file reading, file erasure, or directory reference) is carried out as usual (step S303).

If the file operation request from the user program (100) is a file writing request (step S302), however, the control part (607) judges whether a specific erasure/compression condition has been met based on the free capacity of the storage medium (300) (step S304). In this example, the control part compares the file size of the writing request to the current free capacity of the storage medium (300), and judges that the erasure/compression condition has been met when the size of the free space is less than the file size of the writing request. The control part may also judge that an erasure/compression condition has been met if the current free capacity of the storage medium (300) is less than a specific capacity regardless of the file size of the writing request.

If there is enough free capacity and the erasure/compression condition has not been met (step S305), the requested file is immediately written (step S306). If there is not enough free capacity and the erasure/compression condition has been met (step S305), however, the control part selects the file with the oldest last reference date and time from among uncompressed files, for example, if there are files which may be uncompressed, as Fc, and takes the last reference date and time of this file Fc as t(Fc) (step S307).

Next, the control part selects the file with the oldest last reference date and time from among all of the files being managed as Fd, and takes the last reference date and time of this file Fd as t(Fd) (step S307).

Next, taking the current date and time as t, the control part investigates whether the ratio of |t(Fd) - t| to |t(Fc) - t| is greater than a predetermined constant p; that is, whether the condition of |t(Fd) - t|/|t(Fc) - t| > p has been met (step S309). Fd is erased if this condition has been met; that is, if the file Fd is markedly older than the last reference date and time of the file Fc to be compressed (step S311). If the condition of the step S309 has not been met, however, the file Fc to be compressed is compressed (step S310).

If the last update date and time is used as the erasure or compression attribute, the last update date and time of a file which can be compressed may carry over the last update date and time when uncompressed. If the last reference date and time is used as the erasure or compression attribute, accessing the attributes of a file to be erased or compressed is not viewed as a reference, and does not change the last reference date and time.

Although the first to ninth working examples discussed above did not specify the files to be managed, the files to be managed are not specially limited. For example, files which can be erased or compressed may be limited by using a management system managing, for

example, only the files in the file system of a specific drive, only files beneath a specific directory, or only files used by a specific application. File management also need not necessarily be part of the OS as discussed above, but may be realized as a separate module. Alternately, if the files to be managed are only files used by a specific application, this application may be used to manage these files.

Disseminating the file management method in the first to ninth working examples by storing on recording media such as floppy disks or a CD-ROMs to realize a program which can run on a computer allows the free space of a secondary storage medium such as a disk to be easily managed just by installing this program on a computer from these recording media.

EFFECTS OF THE INVENTION

As discussed above, this invention can provide a computer system with good user friendliness without requiring the user to explicitly select which files should be erased or memorize how to use a program for erasure by an automatic free space assurance processing using file erasure or compression according to the free space on the storage medium managed by the file system. This invention can also store transmitted data arriving externally on the file system without loss of transmitted data and without the user creating a free space. This invention can use file reference history, file correlations, and the number of references to erase files with the lowest probability

of reference by the user, which can improve the utility of automatic erasure.

BRIEF EXPLANATION OF THE DRAWINGS

[Fig. 1] A block diagram showing the functional configuration of a file system used in a computer system according to a first working example of this invention.

[Fig. 2] A diagram showing a file hierarchy handled by the system of this first working example. /17

[Fig. 3] A diagram showing an internal file hierarchy handled by the system of this first working example.

[Fig. 4] A flowchart showing a procedure for automatic file erasure processing in the system of this first working example.

[Fig. 5] A diagram showing an example of a program for making a writing request to the system of this first working example.

[Fig. 6] A diagram illustrating erasure conditions judgement processing used in automatic file erasure processing in the system of this first working example.

[Fig. 7] A flowchart showing a procedure for automatic file erasure processing in the system of this first working example.

[Fig. 8] A block diagram showing the functional configuration of a file system used in a computer system according to a second working example of this invention.

[Fig. 9] A block diagram showing the functional configuration of a file system used in a computer system according to a third working

example of this invention.

[Fig. 10] A flowchart showing a procedure for automatic file erasure processing in the system of this third working example.

[Fig. 11] A diagram showing an example of file importance referenced to select a file to be erased in the system of this second working example.

[Fig. 12] A diagram showing a file system of this first working example realized as a user program independent from the operating system.

[Fig. 13] A block diagram showing the functional configuration of a file system of this first working example when this file system is realized as a user program.

[Fig. 14] A block diagram showing the functional configuration of a file system used in a computer system according to a fourth working example of this invention

[Fig. 15] A diagram showing an example of file reading history information managed by the system of this fourth working example.

[Fig. 16] A diagram showing another example of file reading history information managed by the system of this fourth working example.

[Fig. 17] A diagram showing another example of file reading history information managed by the system of this fourth working example.

[Fig. 18] A flowchart showing a procedure for automatic file erasure processing in the system of this fourth working example.

[Fig. 19] A diagram showing another example of file reading history

information managed by the system of this fourth working example.

[Fig. 20] A flowchart showing another procedure for automatic file erasure processing in the system of this fourth working example.

[Fig. 21] A flowchart showing another procedure for automatic file erasure processing in the system of this fourth working example.

[Fig. 22] A block diagram showing the functional configuration of a file system used in a computer system according to a fifth working example of this invention

[Fig. 23] A flowchart showing a procedure for file reading history information processing in the system of this fifth working example.

[Fig. 24] A file showing file reading history information in the system of this fifth working example.

[Fig. 25] A block diagram showing the functional configuration of a file system used in a computer system according to a sixth working example of this invention

[Fig. 26] A flowchart showing a procedure for automatic file erasure processing in the system of this sixth working example.

[Fig. 27] A block diagram showing the functional configuration of a file system used in a computer system according to a seventh working example of this invention

[Fig. 28] A flowchart showing a procedure for automatic file erasure processing in the system of this seventh working example.

[Fig. 29] A block diagram showing the functional configuration of a file system used in a computer system according to a eighth working

example of this invention

[Fig. 30] A flowchart showing a procedure for automatic file erasure processing in the system of this eighth working example.

[Fig. 31] A block diagram showing the functional configuration of a file system used in a computer system according to a ninth working example of this invention

[Fig. 32] A diagram showing an example of a file management configuration used in the system of this ninth working example.

[Fig. 33] A flowchart showing a procedure for free space assurance processing selectively using automatic file erasure processing and automatic file compression in the system of this ninth working example.

[Fig. 34] A flowchart showing another procedure for free space assurance processing selectively using automatic file erasure processing and automatic file compression in the system of this ninth working example.

EXPLANATION OF REFERENCE NUMERALS

1, 10 … user program, 2, 20 … file system, 3, 30 … storage medium, 4 … notification program, 5 … operating system, 6 … file management program, 21 … request reception part, 22 … erasure/compression conditions judgment part, 23 … file erasure/compression selection part, 24 … file erasure/compression part, 25 … file input/output part, 26 … erasure/compression notification part, 27 … permission information reception part, 32 … file reading history storage part,

33 … file size information storage part, 34 … file correlation storage part, 201 … file input/output part, 202 … file reading part, 203 … file reading history creation part, 204 … fluorochemical adhesive material selection part, 205 file erasure part, 206 … file writing part, 303 … file size information creation part, 403 … file correlation creation part, 502 … reference correlation reading part

FIG. 1

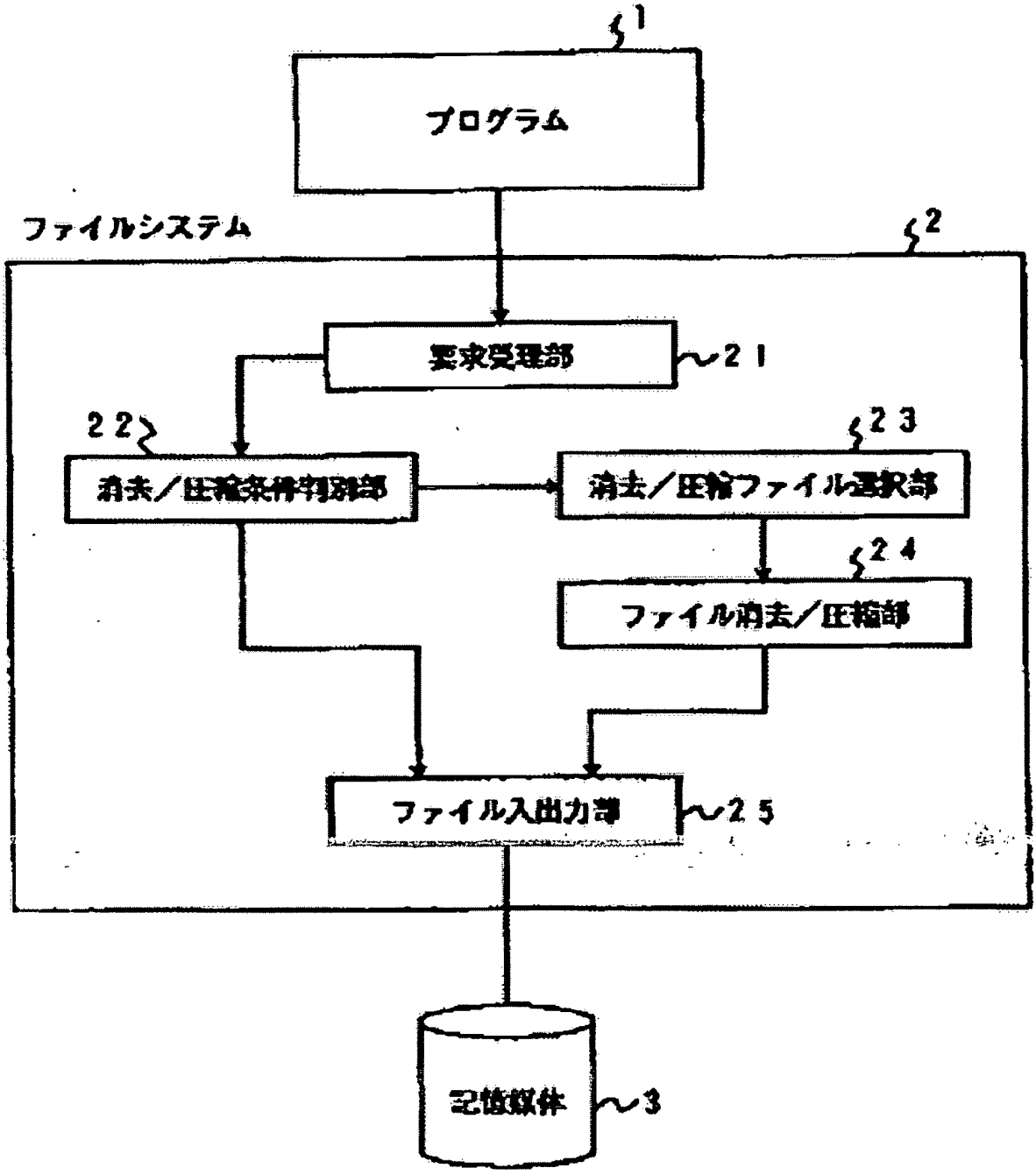

1
プログラム
ファイルシステム
2
要求受理部
21
22
消去／圧縮条件判別部
23
消去／圧縮ファイル選択部
24
ファイル消去／圧縮部
ファイル入出力部
25
記憶媒体
3

FIG. 2

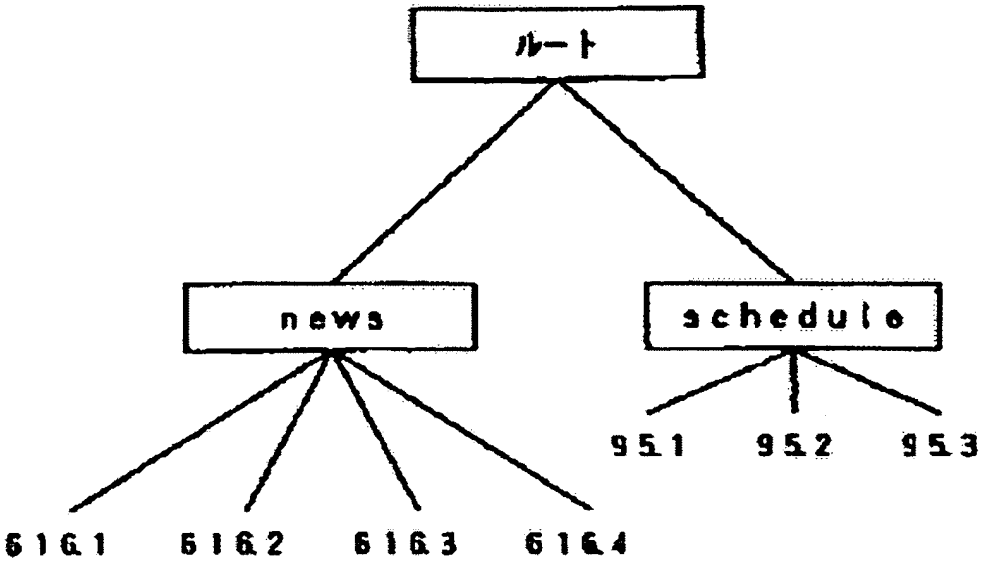

ルート
news
schedule
616.1
616.2
616.3
616.4
95.1
95.2
95.3

FIG. 3

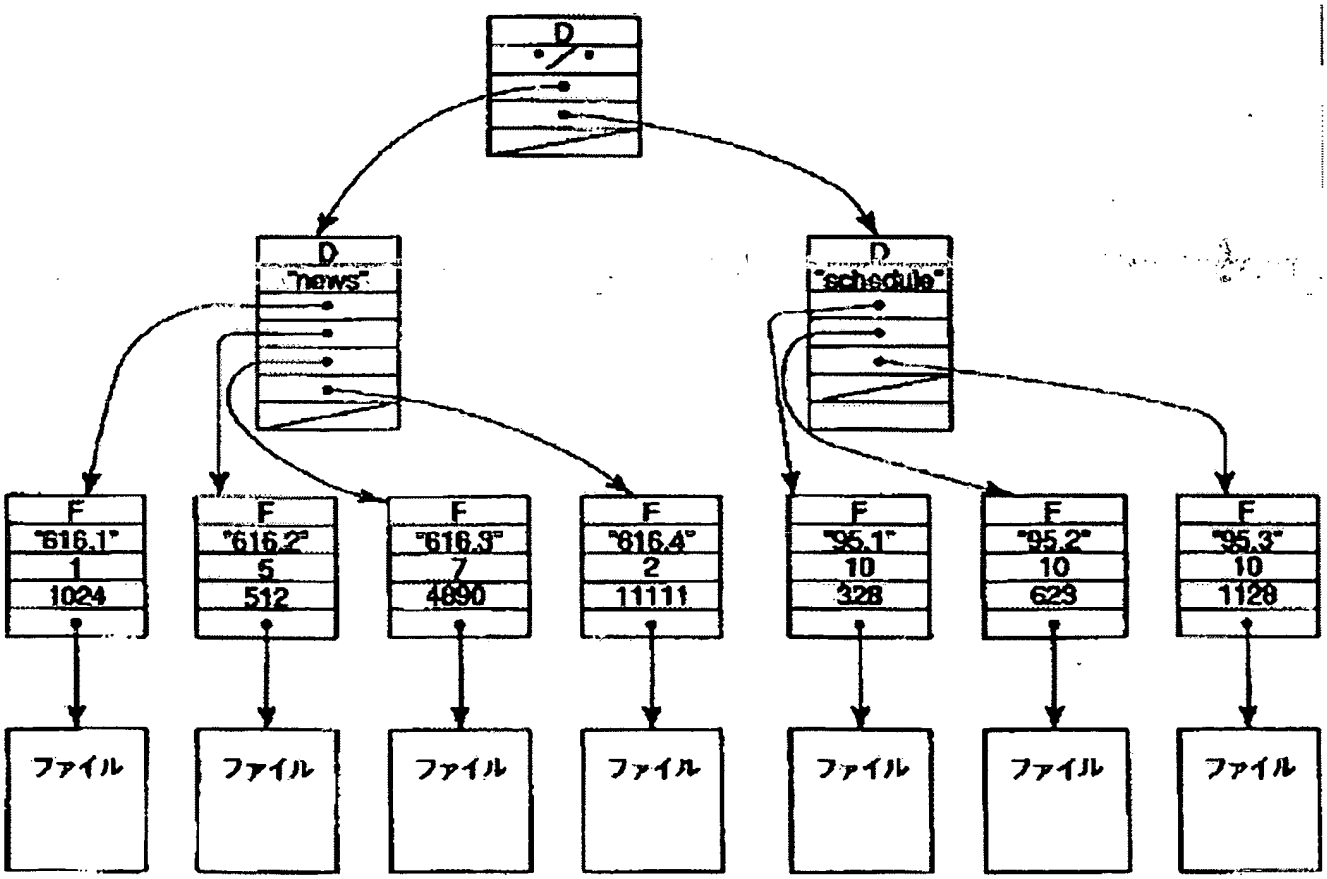

D
D
"news"
D
"schedule"
F
"616.1"
1
1024
F
"616.2"
5
512
F
"616.3"
7
4890
F
"616.4"
2
11111
F
"95.1"
10
328
F
"95.2"
10
623
F
"95.3"
10
1128
ファイル
ファイル
ファイル
ファイル
ファイル
ファイル
ファイル

FIG. 4

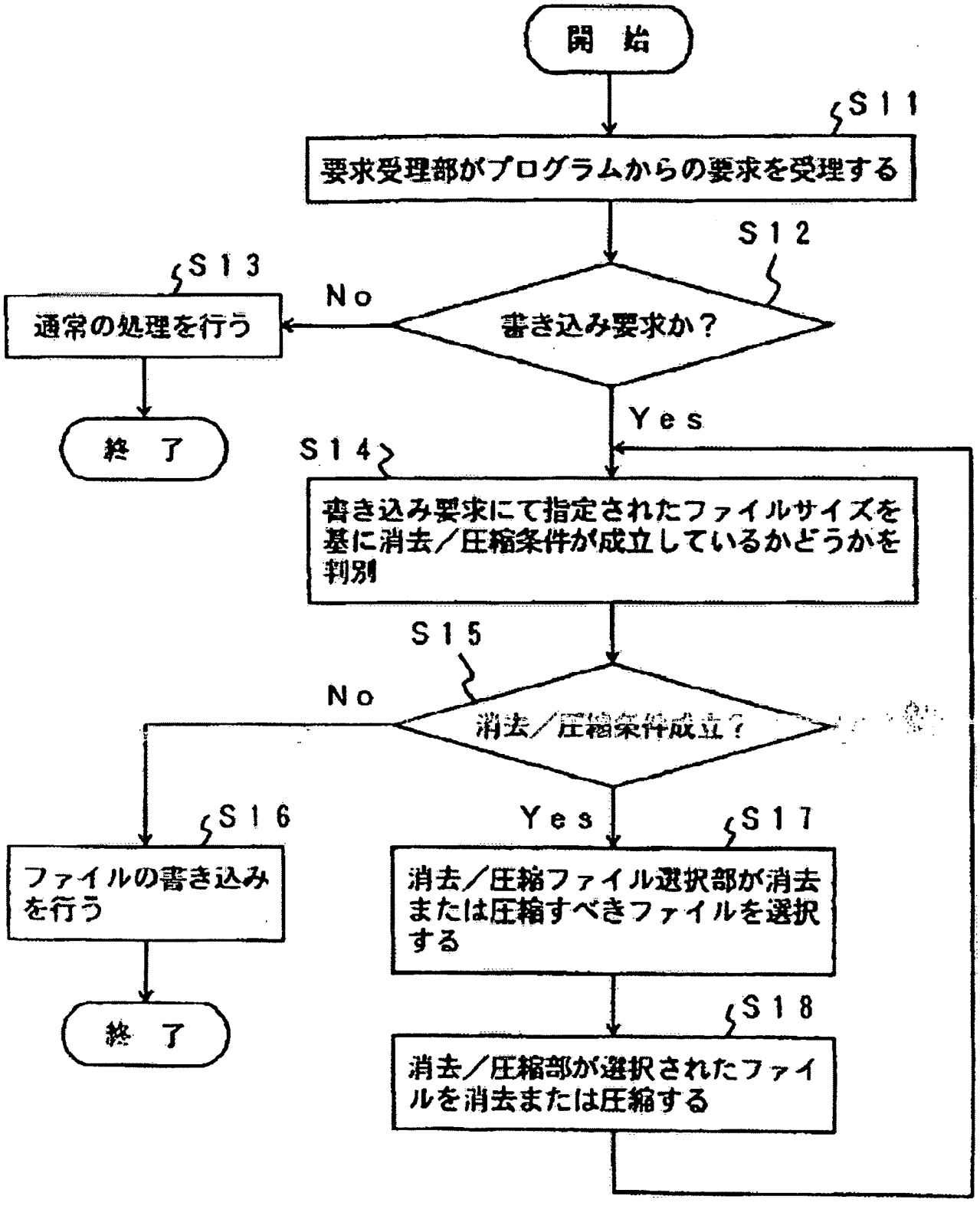

開始
S11
要求受理部がプログラムからの要求を受理する
S12
書き込み要求か？
No
S13
通常の処理を行う
終了
Yes
S14
書き込み要求にて指定されたファイルサイズを基に消去／圧縮条件が成立しているかどうかを判別
S15
消去／圧縮条件成立？
No
S16
ファイルの書き込みを行う
終了
Yes
S17
消去／圧縮ファイル選択部が消去または圧縮すべきファイルを選択する
S18
消去／圧縮部が選択されたファイルを消去または圧縮する

FIG. 5

```
fd=creat("/schedule/956") : (1)

rval=write(fd, buf, 256) ;     (2)
```

FIG. 6

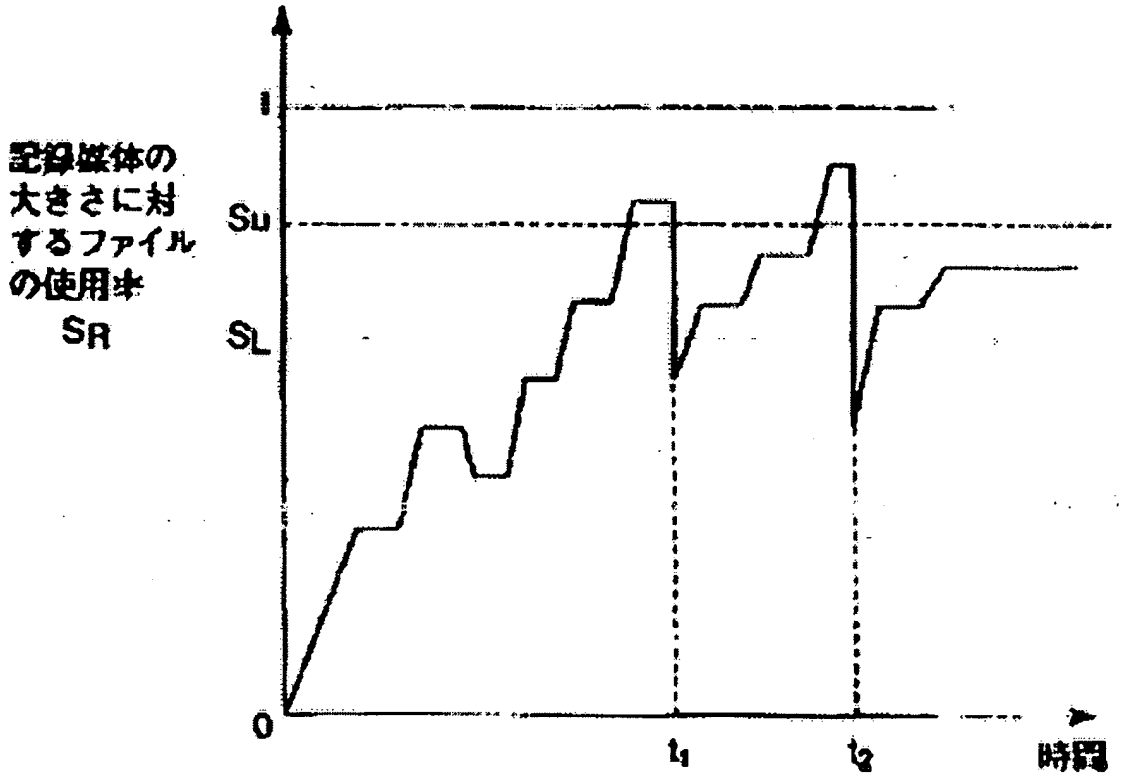


FIG. 7

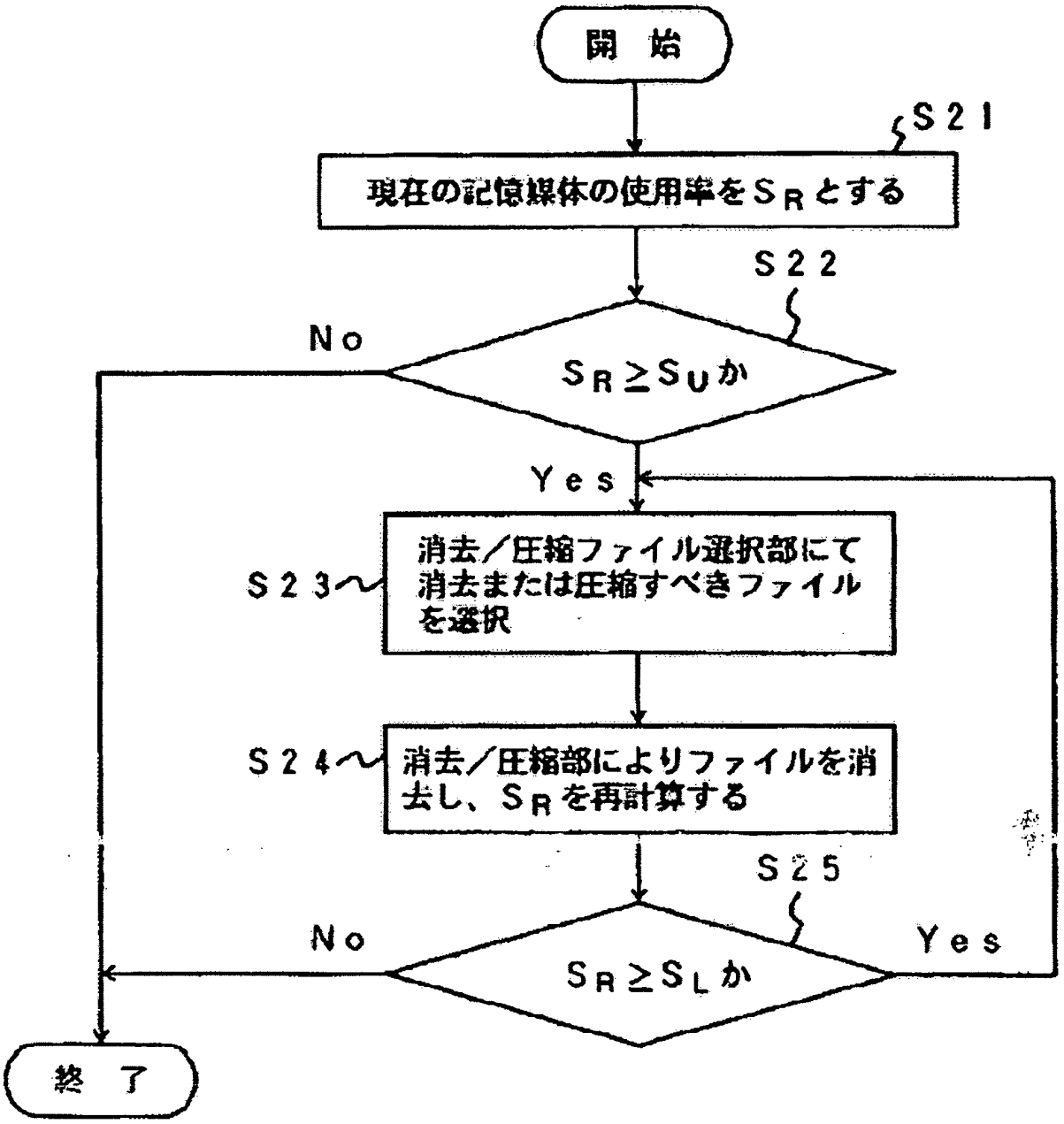

開 始
S21
現在の記憶媒体の使用率をSRとする
S22
No
SR≧SUか
Yes
S23
消去／圧縮ファイル選択部にて消去または圧縮すべきファイルを選択
S24
消去／圧縮部によりファイルを消去し、SRを再計算する
S25
No
SR≧SLか
Yes
終 了

FIG. 8

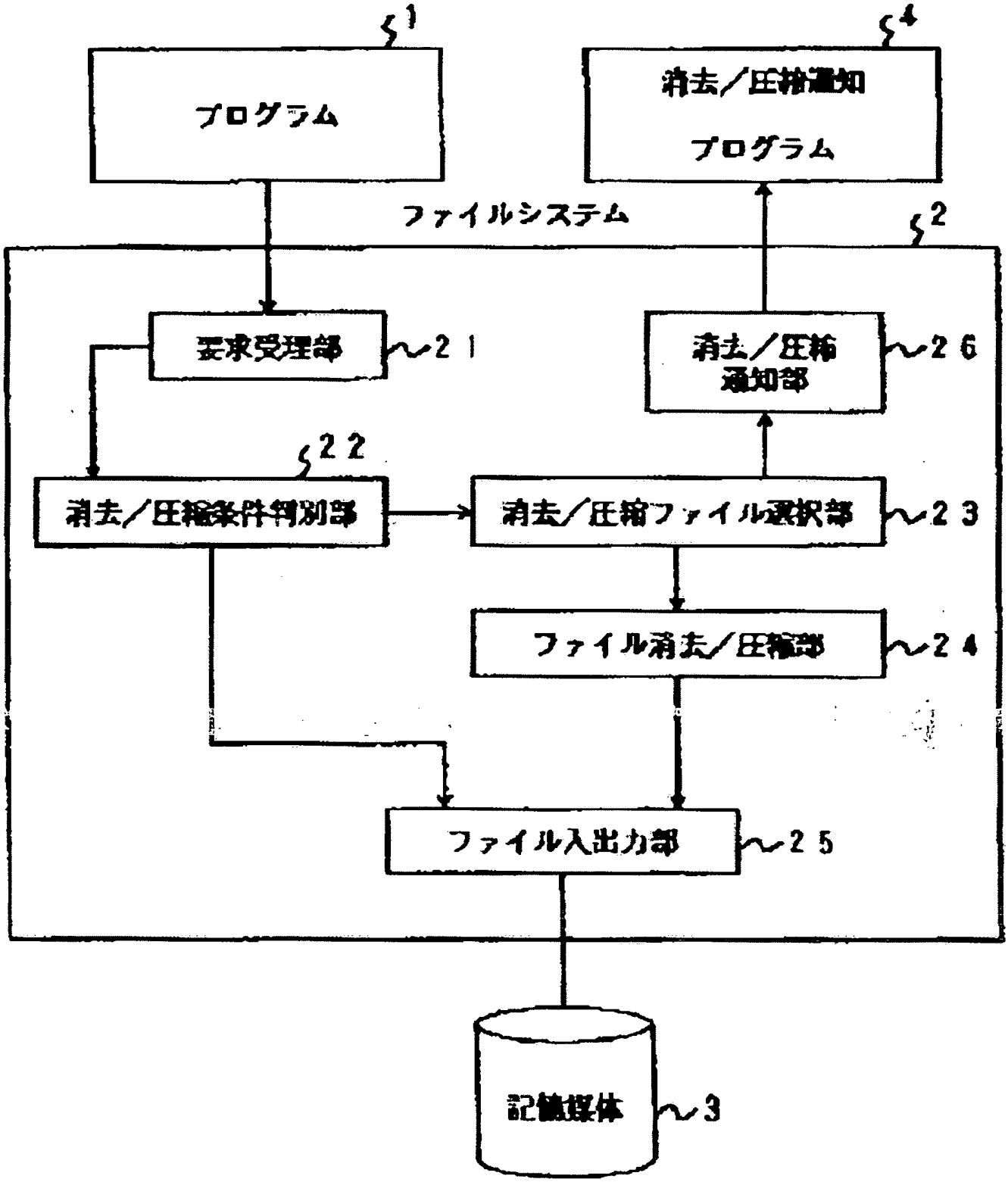

1
プログラム
4
消去／圧縮通知
プログラム
ファイルシステム
2
要求受理部
21
消去／圧縮
通知部
26
22
消去／圧縮条件判別部
消去／圧縮ファイル選択部
23
ファイル消去／圧縮部
24
ファイル入出力部
25
記憶媒体
3

FIG. 9

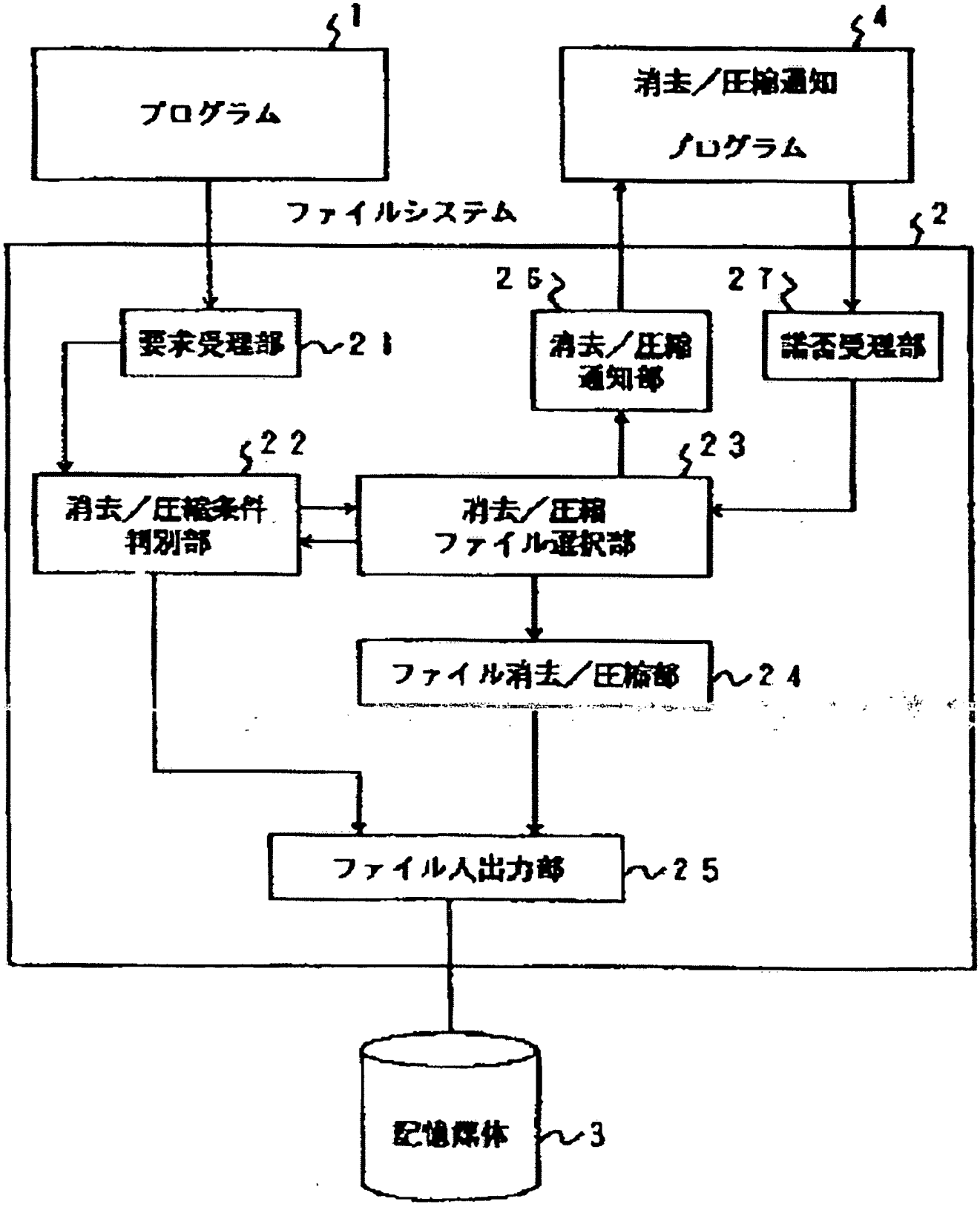

1
プログラム
4
消去／圧縮通知
プログラム
ファイルシステム
2
25
27
要求受理部
21
消去／圧縮
通知部
諾否受理部
22
23
消去／圧縮条件
判別部
消去／圧縮
ファイル選択部
ファイル消去／圧縮部
24
ファイル入出力部
25
記憶媒体
3

FIG. 10

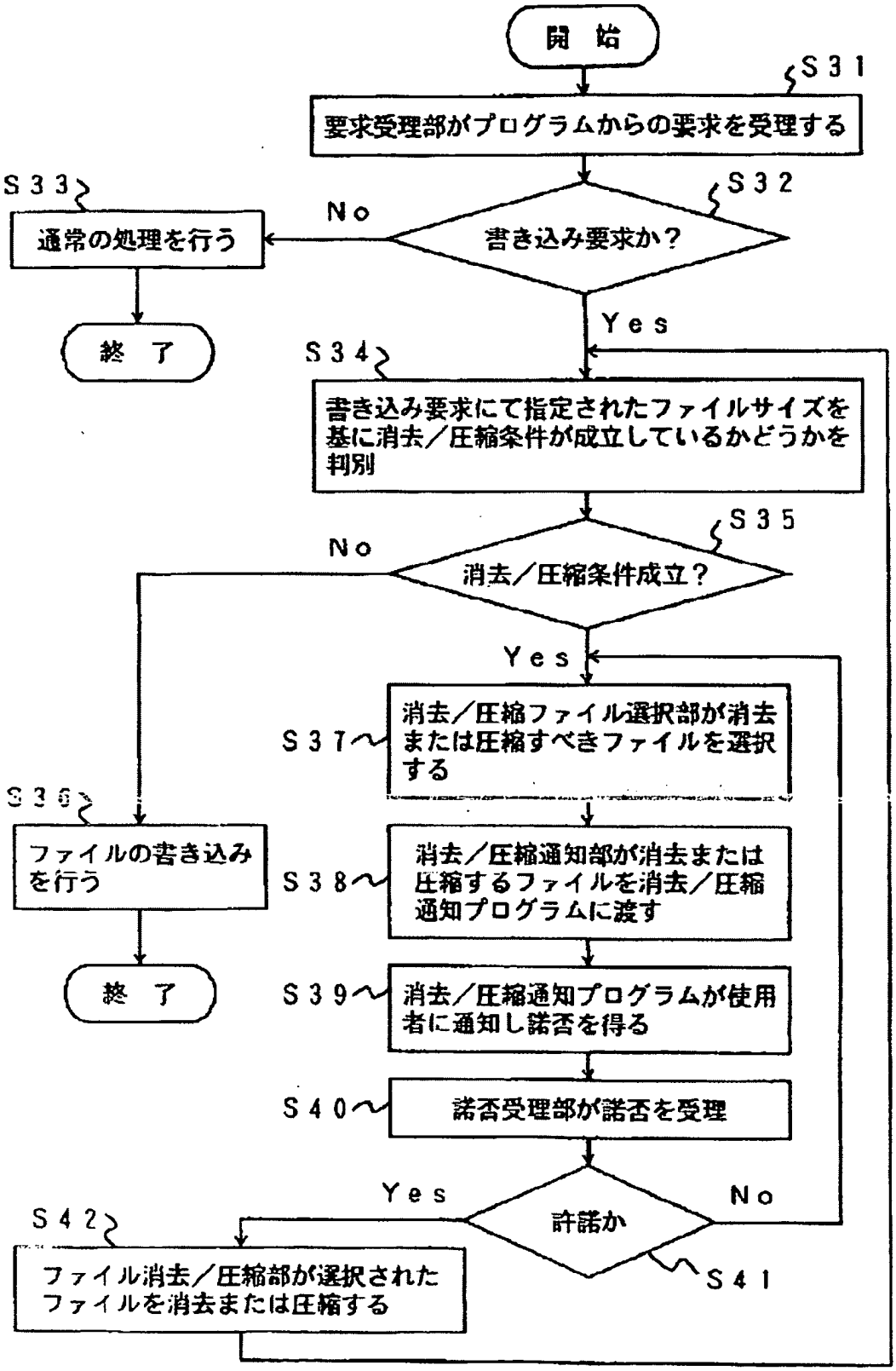

開 始
S31
要求受理部がプログラムからの要求を受理する
S32
書き込み要求か？
No
S33
通常の処理を行う
終 了
Yes
S34
書き込み要求にて指定されたファイルサイズを基に消去／圧縮条件が成立しているかどうかを判別
S35
消去／圧縮条件成立？
No
Yes
S36
ファイルの書き込みを行う
終 了
S37
消去／圧縮ファイル選択部が消去または圧縮すべきファイルを選択する
S38
消去／圧縮通知部が消去または圧縮するファイルを消去／圧縮通知プログラムに渡す
S39
消去／圧縮通知プログラムが使用者に通知し諾否を得る
S40
諾否受理部が諾否を受理
許諾か
Yes
No
S41
S42
ファイル消去／圧縮部が選択されたファイルを消去または圧縮する

FIG. 11

/news/616.1	1
/news/616.2	5
/news/616.3	7
/news/616.4	2
/schedule/951	10
/schedule/952	10
/schedule/953	10

FIG. 12

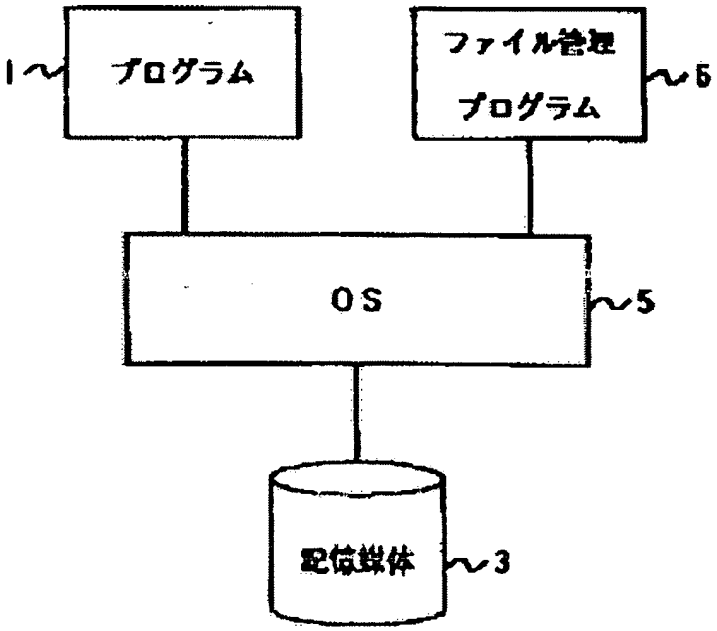


FIG. 13

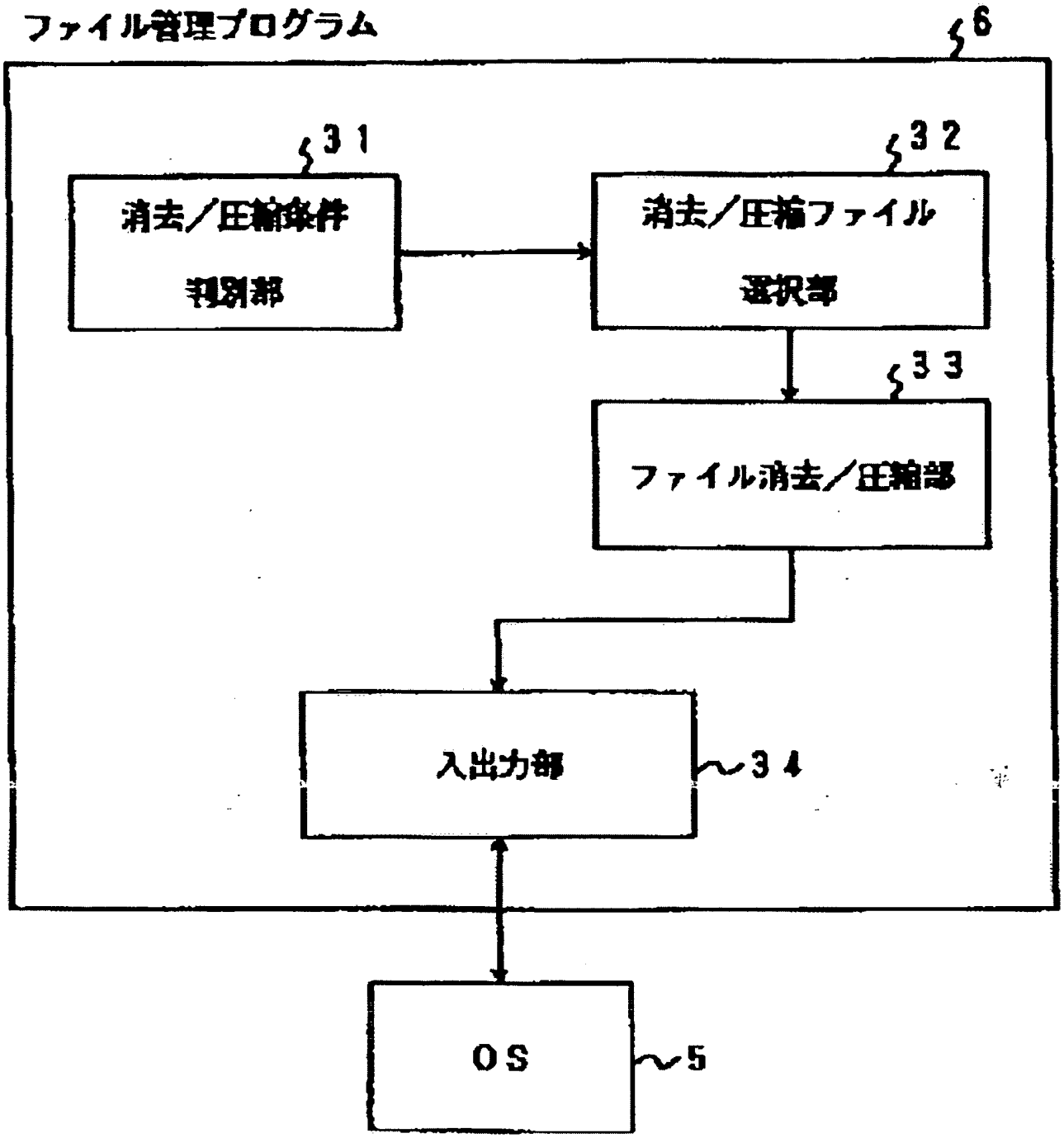

ファイル管理プログラム
6
31
消去／圧縮条件
判別部
32
消去／圧縮ファイル
選択部
33
ファイル消去／圧縮部
入出力部
34
OS
5

FIG. 14

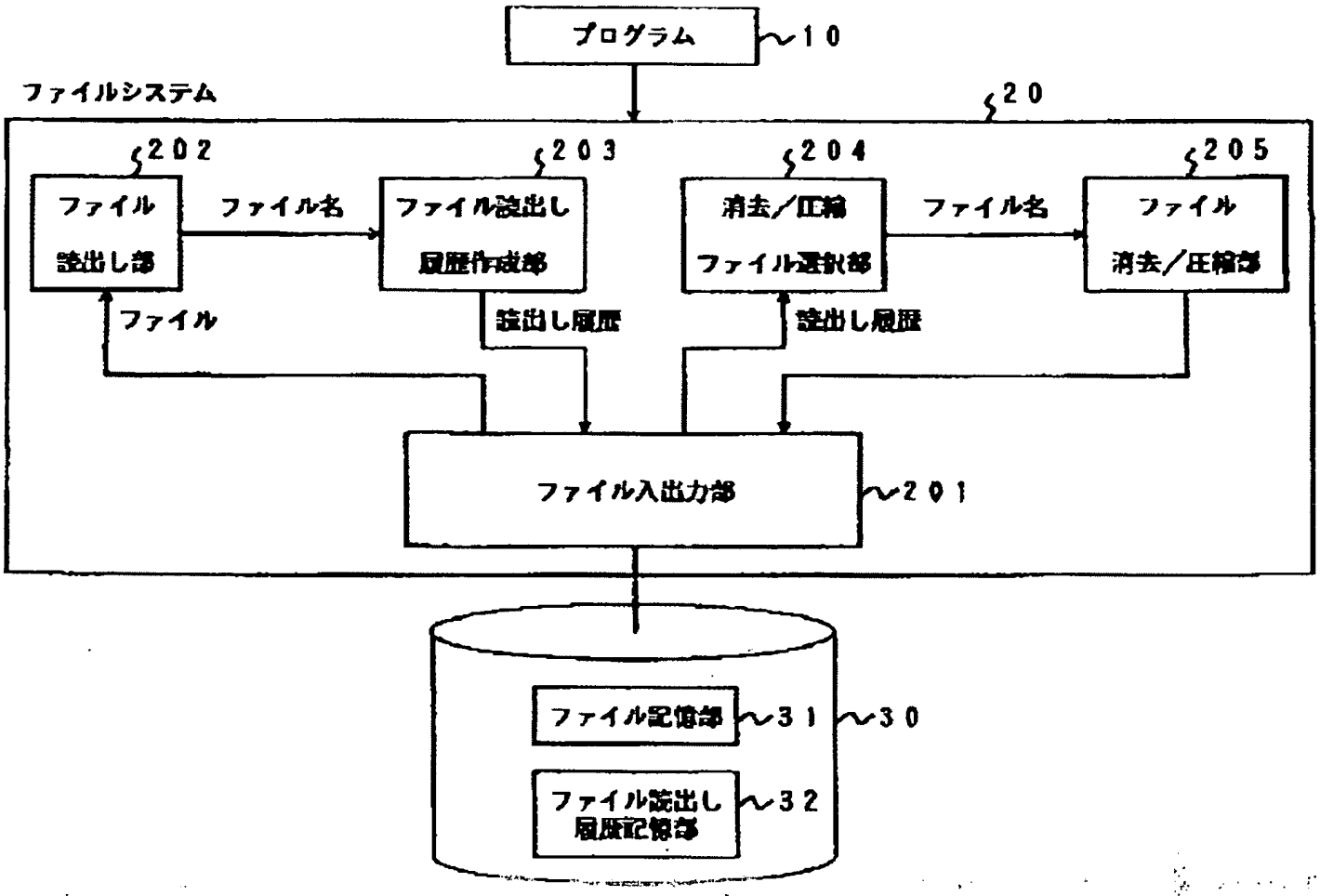

プログラム
10
ファイルシステム
20
202
203
204
205
ファイル
読出し部
ファイル名
ファイル読出し
履歴作成部
消去／圧縮
ファイル選択部
ファイル名
ファイル
消去／圧縮部
ファイル
読出し履歴
読出し履歴
ファイル入出力部
201
ファイル記憶部
31
30
ファイル読出し
履歴記憶部
32

FIG. 15

ディレクトリ

ファイル名	ファイル タイプ	----	読出し履歴 (読出しの有無)
f 1			0
f 2			1
f 3			1
f 4			0
⋮	⋮	⋮	⋮

0 : 読出されたことのないファイル

1 : 読出されたことのあるファイル

FIG. 16

ディレクトリ

ファイル名	ファイル タイプ	----	読出し履歴 (読出し回数)
f 1			10
f 2			0
f 3			20
f 4			5
⋮	⋮	⋮	⋮

FIG. 17

ディレクトリ

ファイル名	ファイルタイプ	-------
⋮	⋮	⋮

(A)

読出し履歴ファイル

ファイル名	読出し履歴
⋮	⋮

(B)

FIG. 18

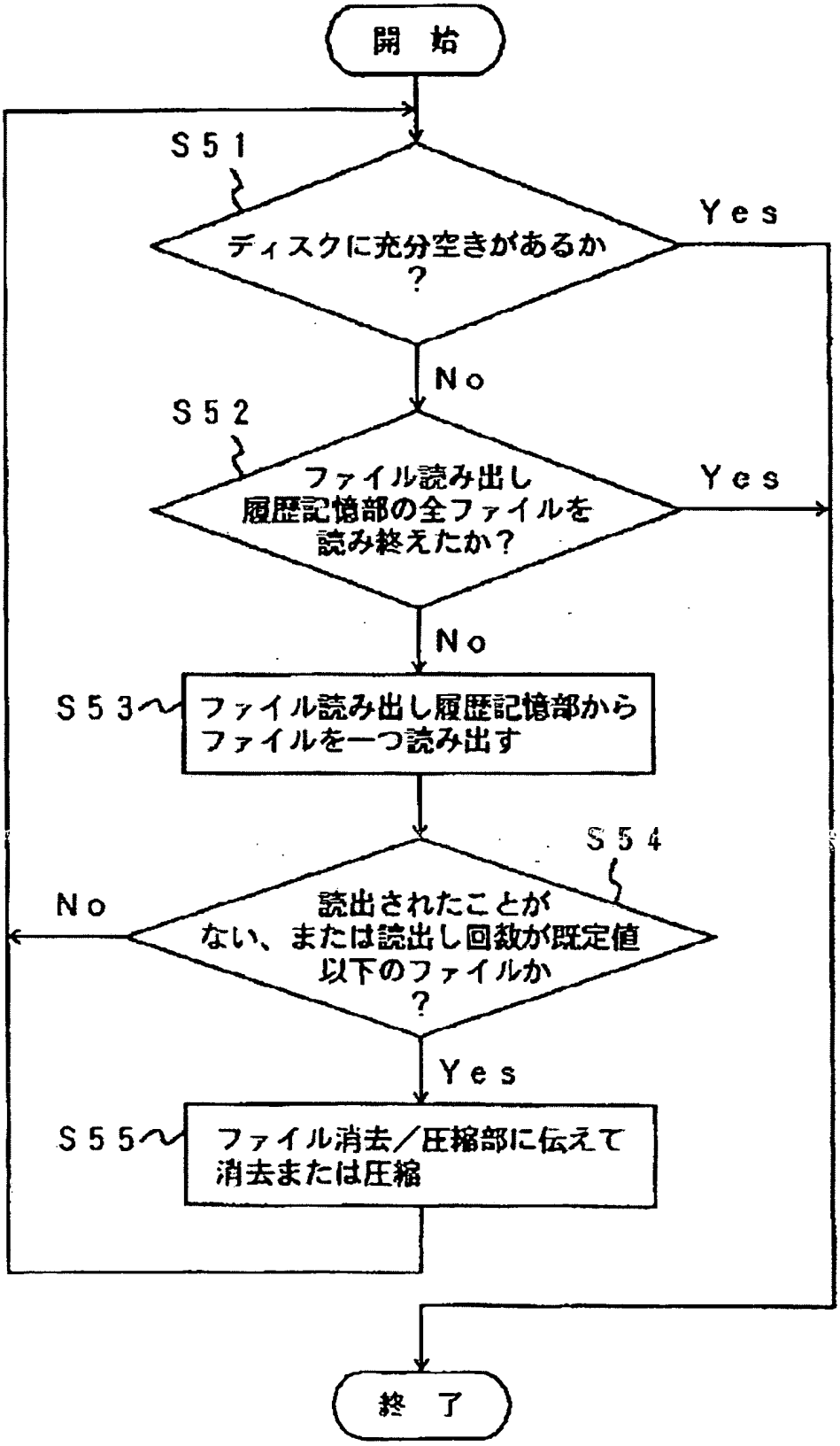

開 始
S51
ディスクに充分空きがあるか？
Yes
No
S52
ファイル読み出し履歴記憶部の全ファイルを読み終えたか？
Yes
No
S53
ファイル読み出し履歴記憶部からファイルを一つ読み出す
S54
読出されたことがない、または読出し回数が既定値以下のファイルか？
No
Yes
S55
ファイル消去／圧縮部に伝えて消去または圧縮
終 了

FIG. 19

読出し履歴ファイル

ファイル名	読出し履歴 (最終読出し日時)	
f 1	1 0 : 3 0	9 5 − 8 − 7
f 2	1 7 : 3 0	9 5 − 8 − 6
f 3	1 0 : 3 0	9 5 − 8 − 6
f 4	1 7 : 3 0	9 5 − 8 − 5
f 5	1 0 : 3 0	9 5 − 8 − 5
⋮	⋮	

FIG. 20

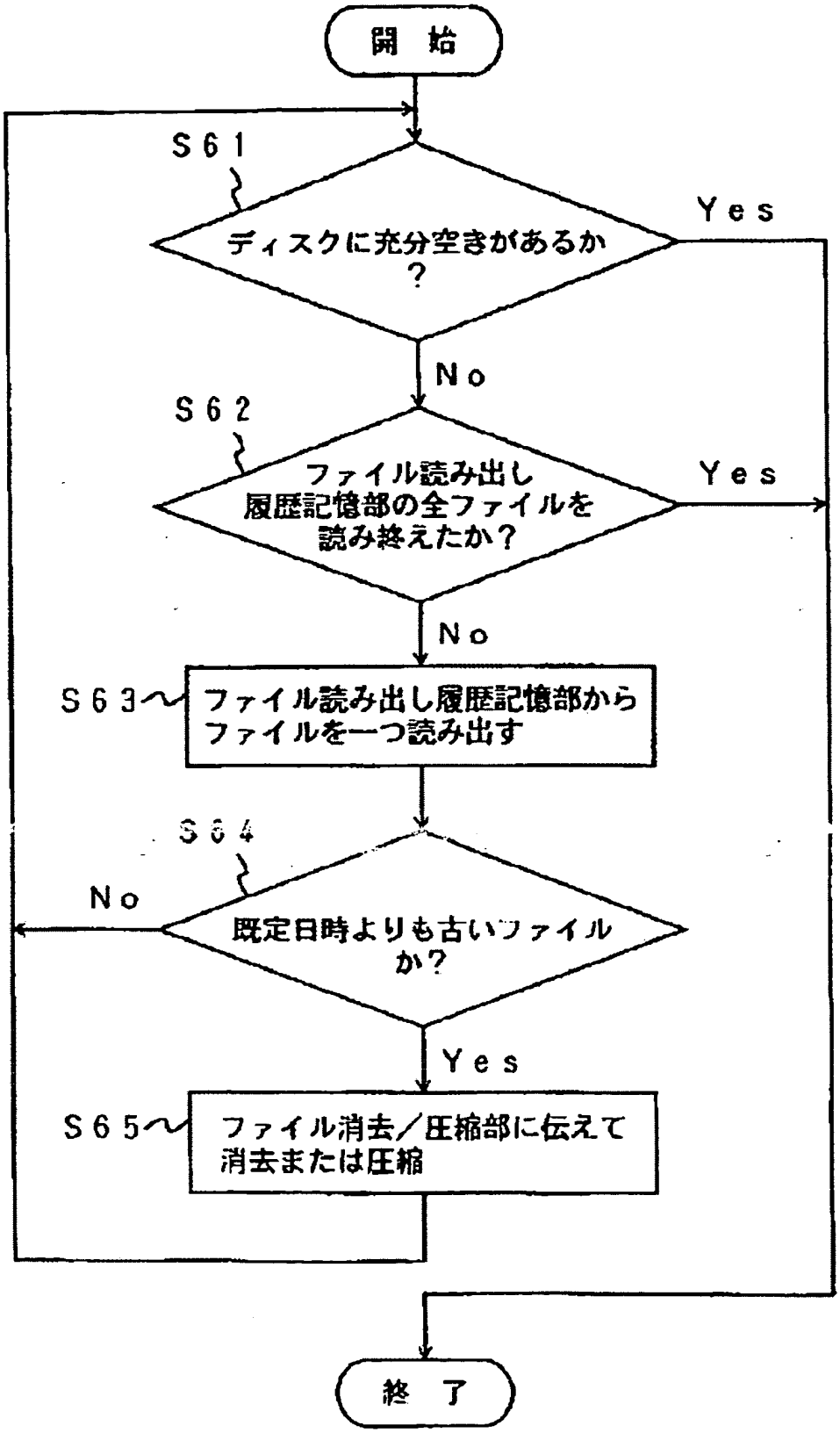

開 始
S61
ディスクに充分空きがあるか？
Yes
No
S62
ファイル読み出し履歴記憶部の全ファイルを読み終えたか？
Yes
No
S63
ファイル読み出し履歴記憶部からファイルを一つ読み出す
S64
No
既定日時よりも古いファイルか？
Yes
S65
ファイル消去／圧縮部に伝えて消去または圧縮
終 了

FIG. 21

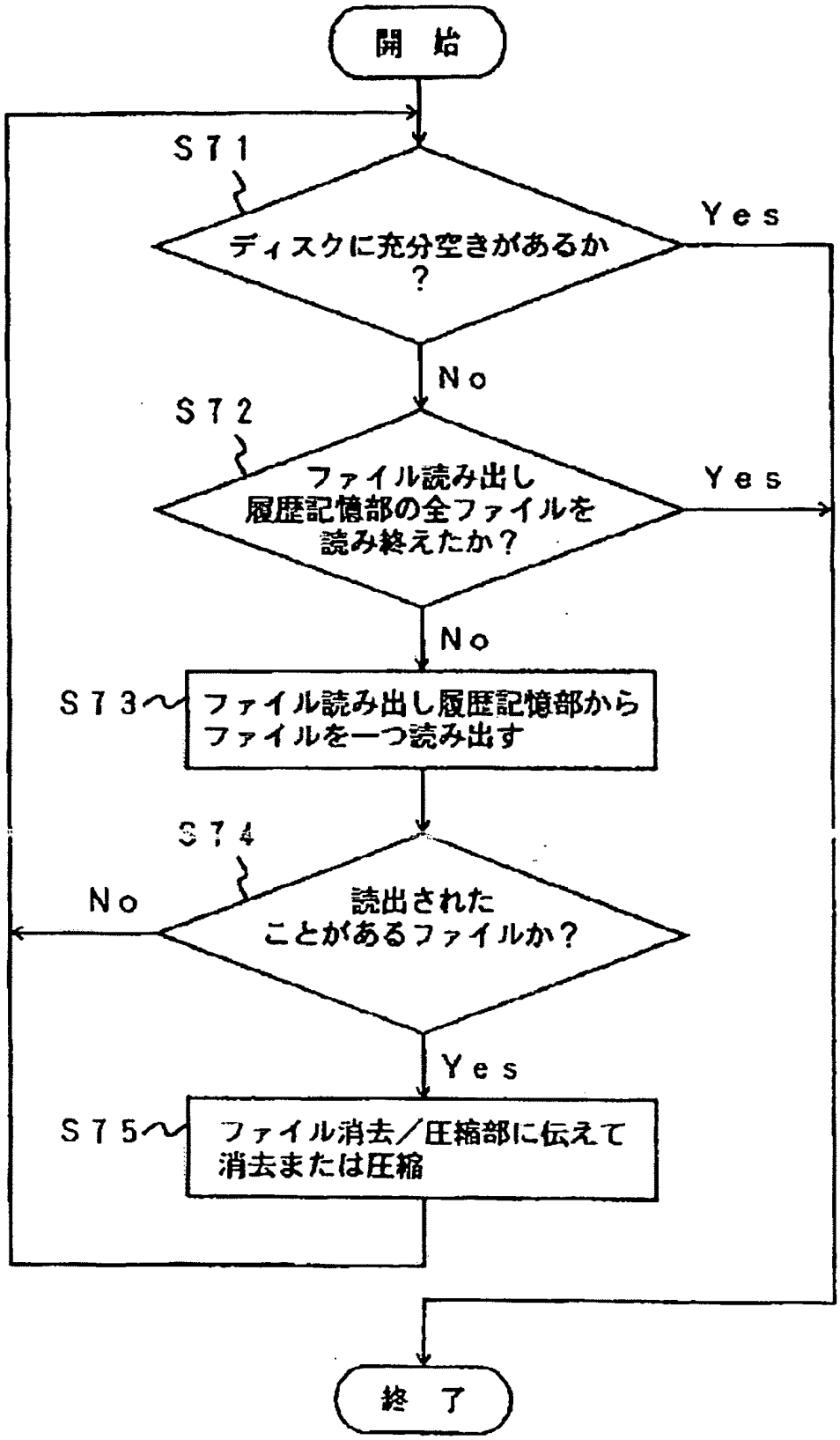

開 始
S71
ディスクに充分空きがあるか？
Yes
No
S72
ファイル読み出し履歴記憶部の全ファイルを読み終えたか？
Yes
No
S73
ファイル読み出し履歴記憶部からファイルを一つ読み出す
S74
No
読出されたことがあるファイルか？
Yes
S75
ファイル消去／圧縮部に伝えて消去または圧縮
終 了

FIG. 22

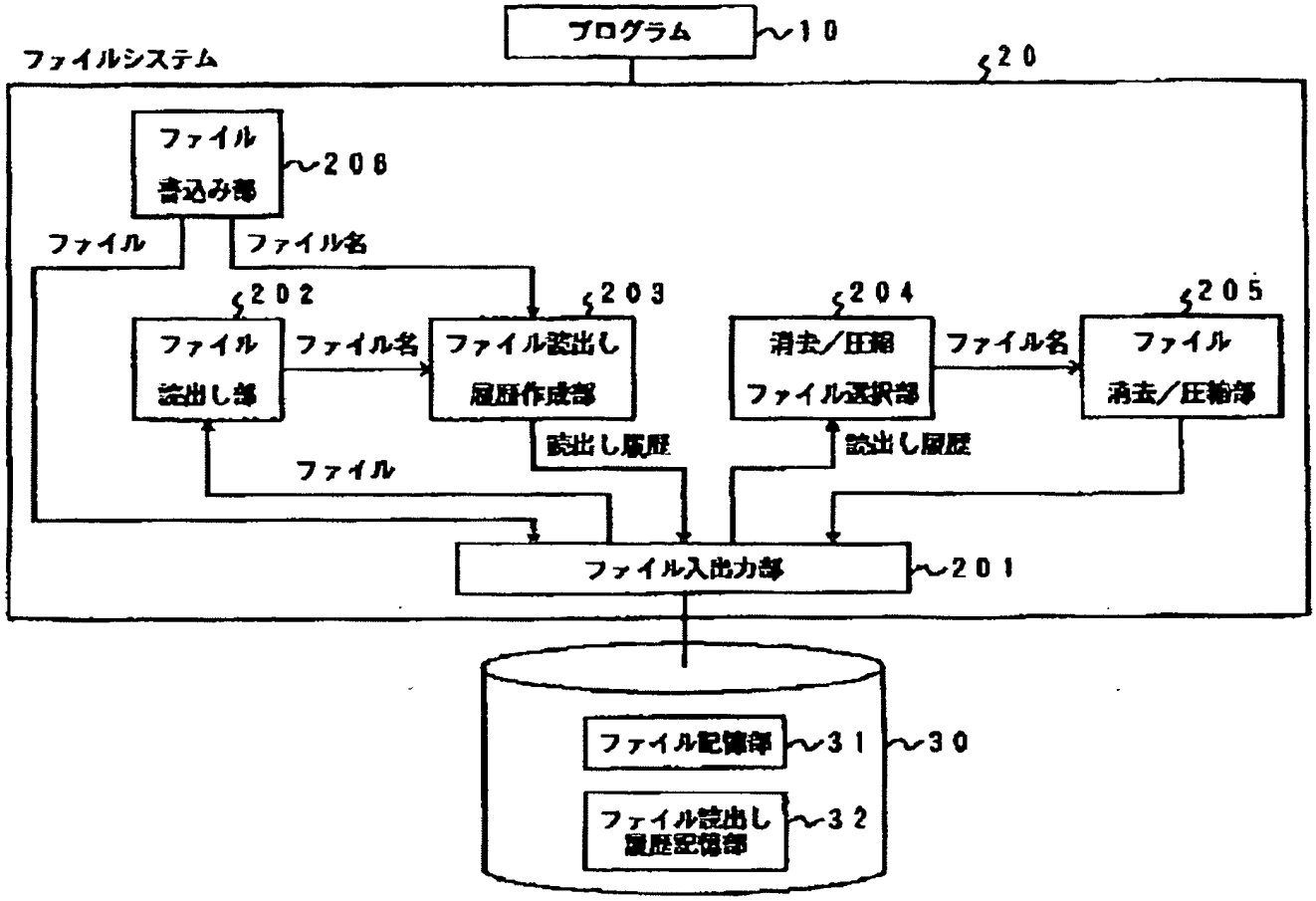

プログラム
10
ファイルシステム
20
ファイル
書込み部
206
ファイル
ファイル名
202
203
204
205
ファイル
読出し部
ファイル名
ファイル読出し
履歴作成部
消去/圧縮
ファイル選択部
ファイル名
ファイル
消去/圧縮部
読出し履歴
読出し履歴
ファイル
ファイル入出力部
201
ファイル記憶部
31
30
ファイル読出し
履歴記憶部
32

FIG. 23

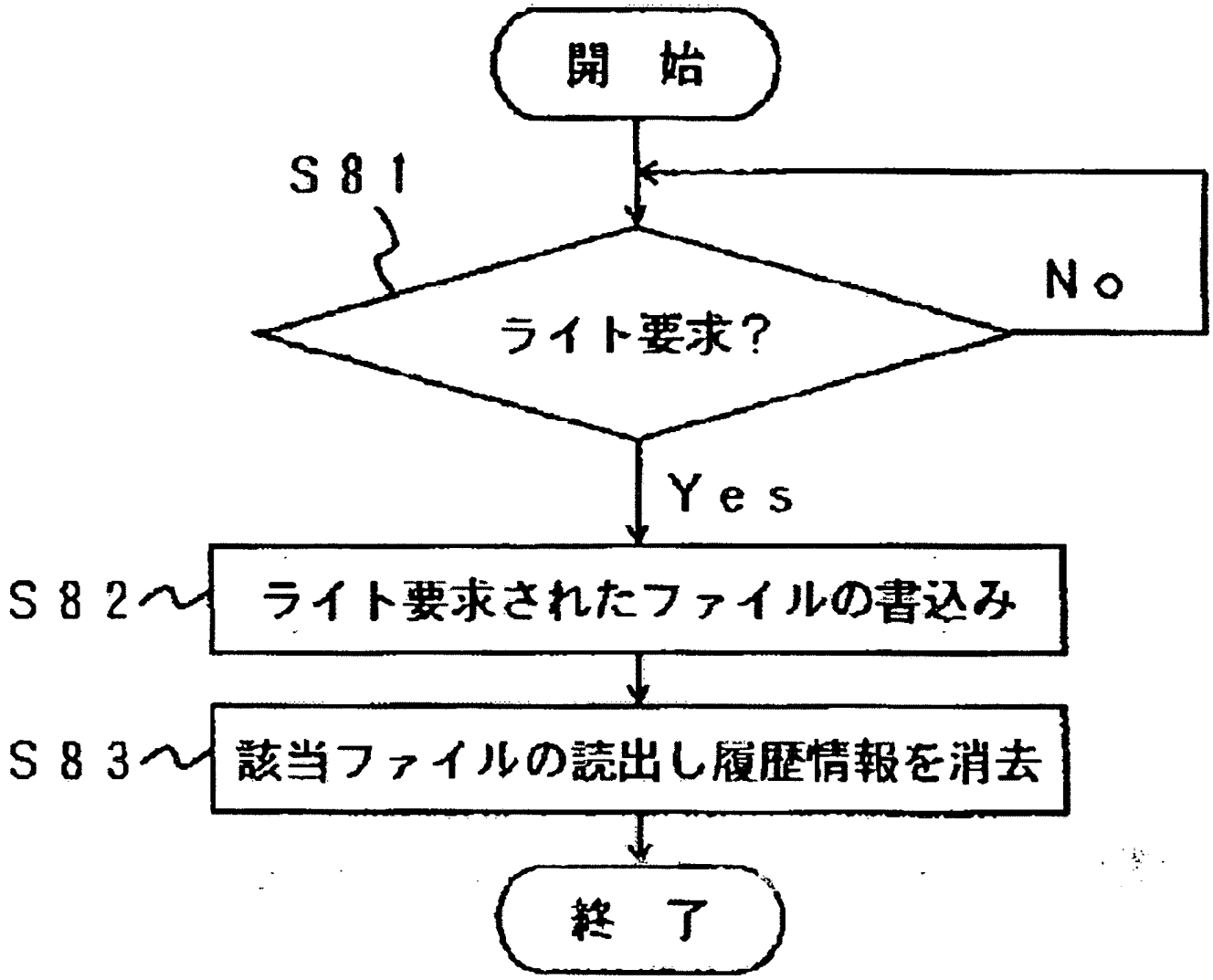

開　始
S81
ライト要求？
No
Yes
S82
ライト要求されたファイルの書込み
S83
該当ファイルの読出し履歴情報を消去
終　了

FIG. 24

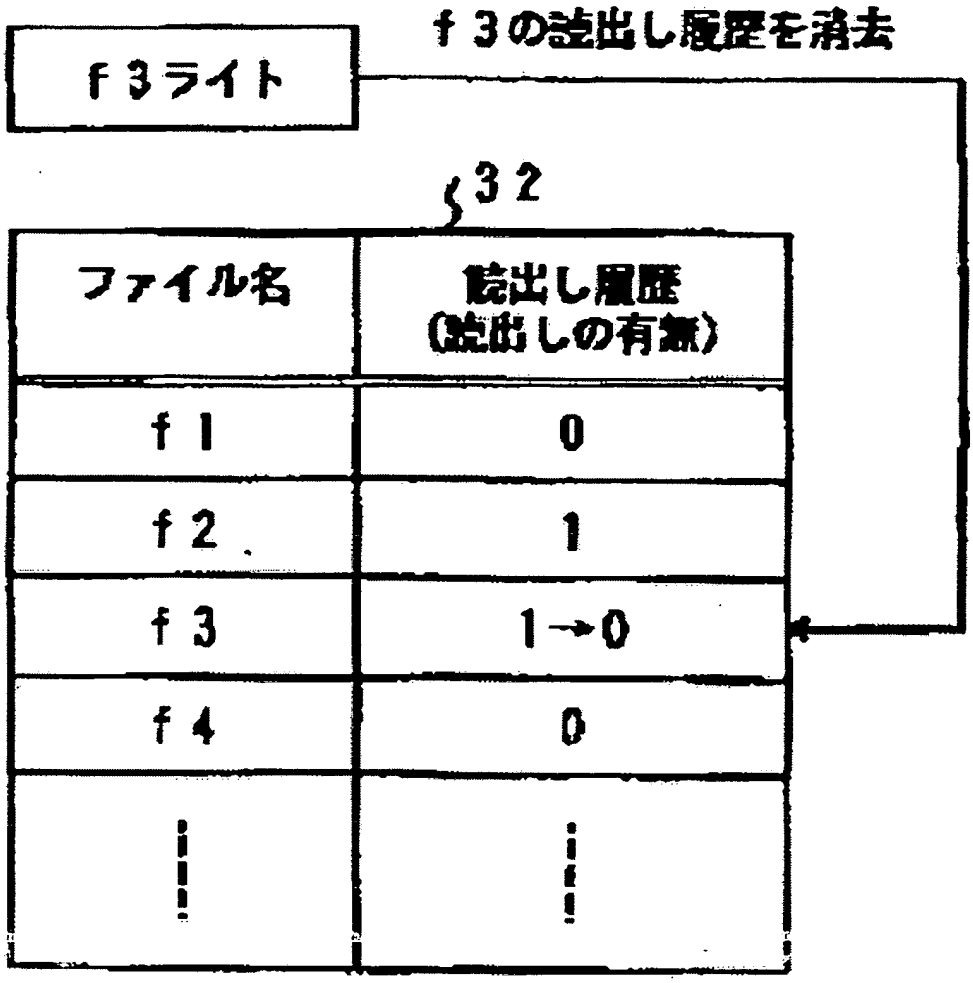


FIG. 25

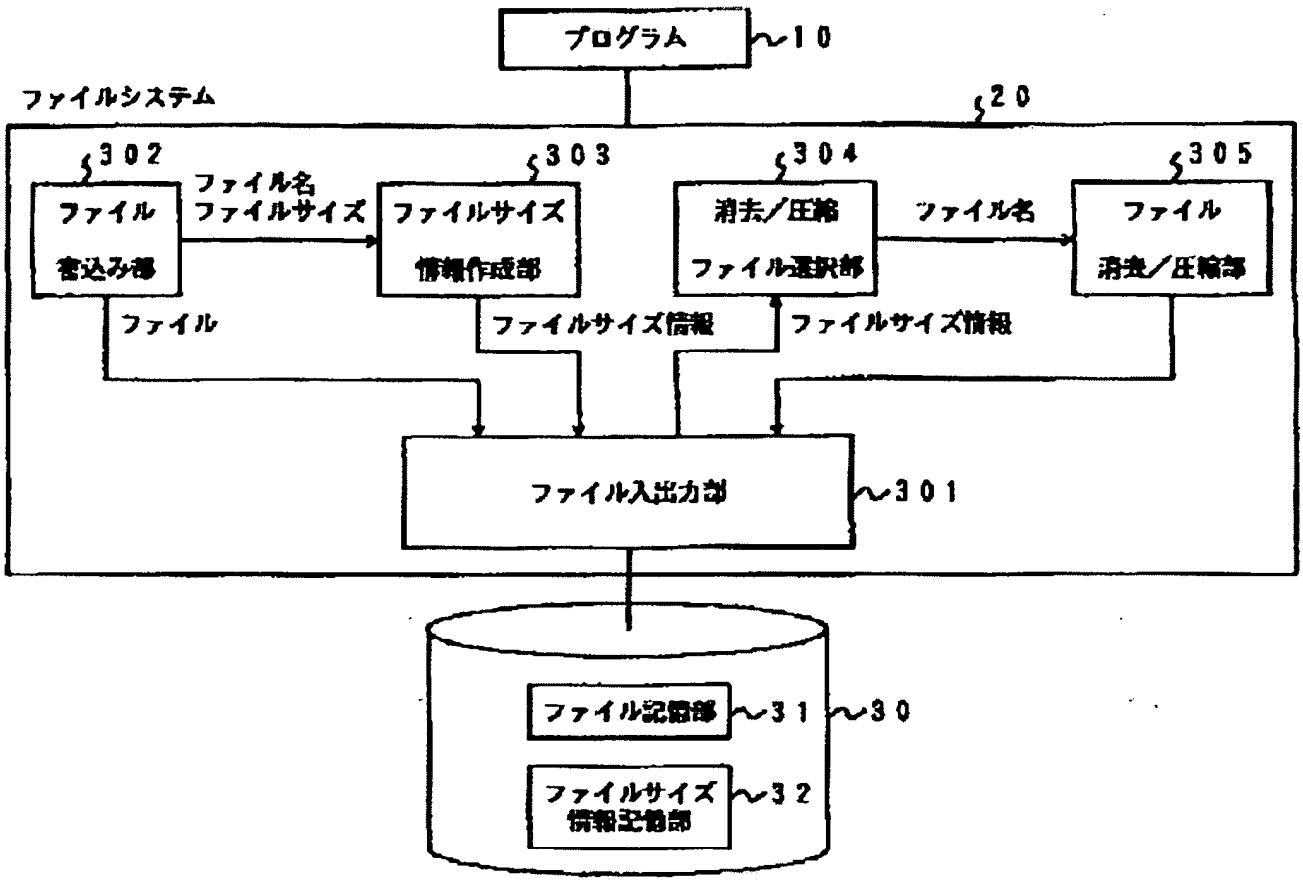

プログラム
10
ファイルシステム
20
302
ファイル
書込み部
ファイル名
ファイルサイズ
303
ファイルサイズ
情報作成部
304
消去／圧縮
ファイル選択部
ファイル名
305
ファイル
消去／圧縮部
ファイル
ファイルサイズ情報
ファイルサイズ情報
ファイル入出力部
301
ファイル記憶部
31
30
ファイルサイズ
情報記憶部
32

FIG. 26

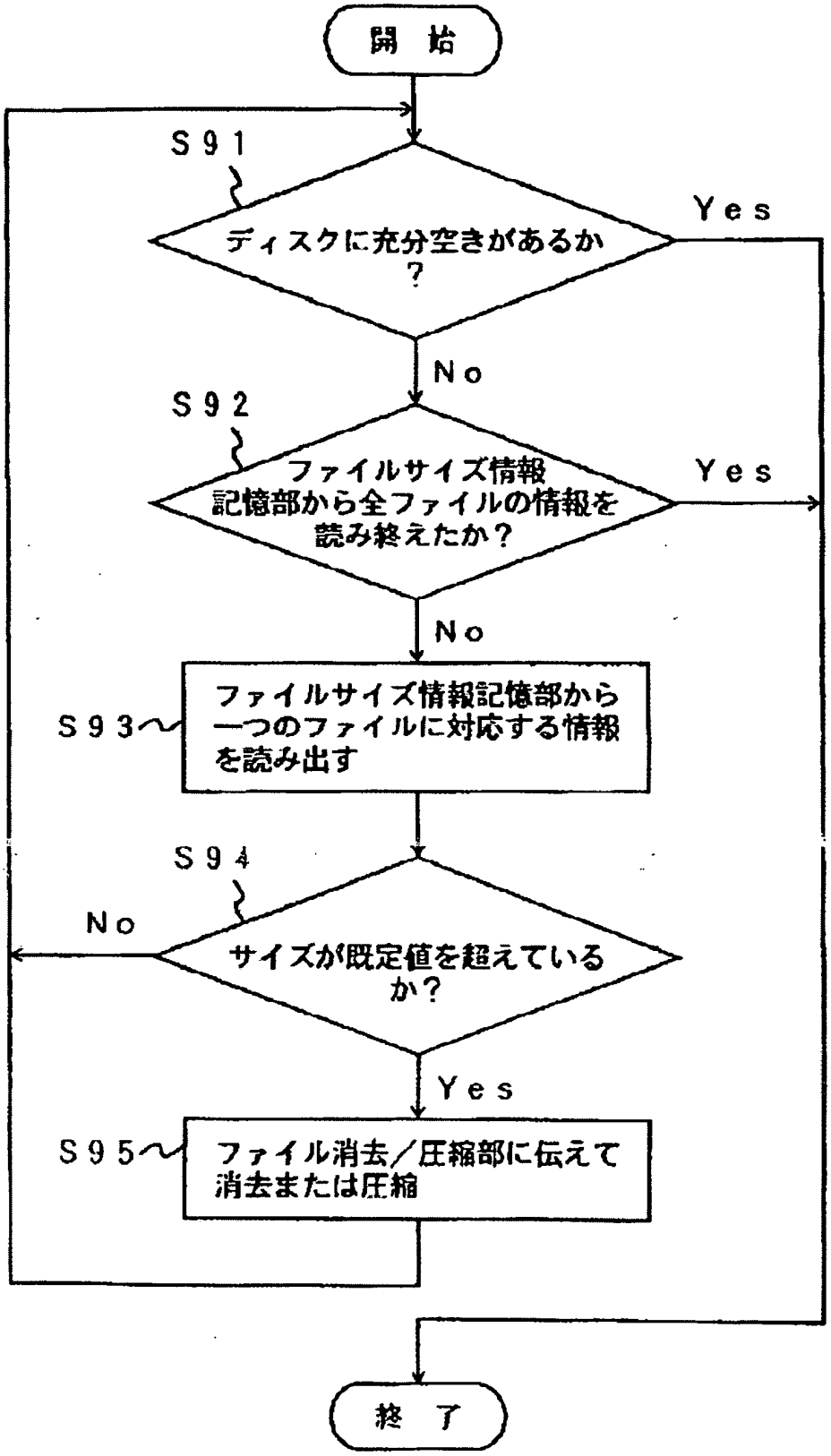

開 始
S91
ディスクに充分空きがあるか？
Yes
No
S92
ファイルサイズ情報記憶部から全ファイルの情報を読み終えたか？
Yes
No
S93
ファイルサイズ情報記憶部から一つのファイルに対応する情報を読み出す
S94
No
サイズが既定値を超えているか？
Yes
S95
ファイル消去／圧縮部に伝えて消去または圧縮
終 了

FIG. 27

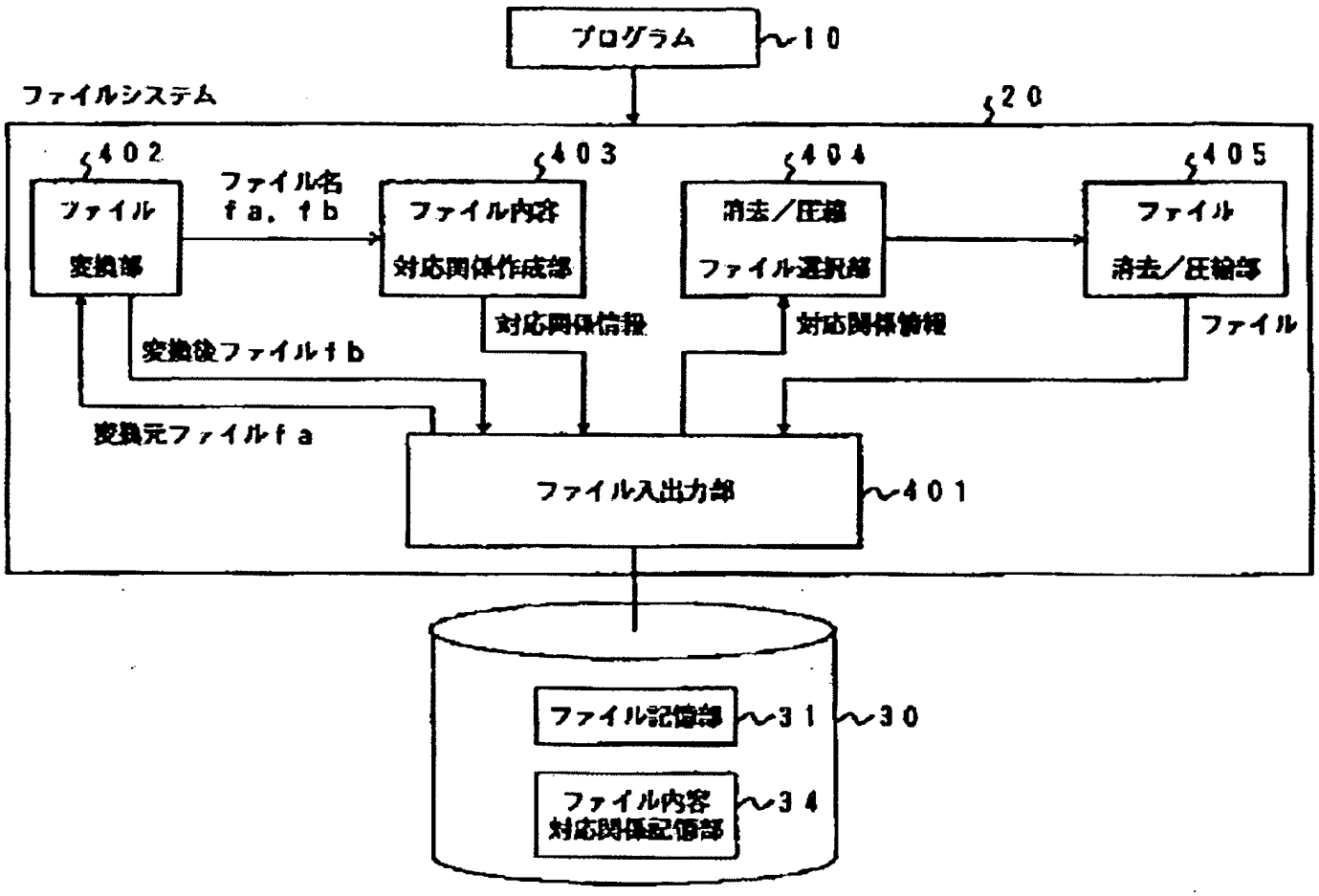

プログラム
10
ファイルシステム
20
402
ファイル
変換部
ファイル名
fa, fb
403
ファイル内容
対応関係作成部
404
消去／圧縮
ファイル選択部
405
ファイル
消去／圧縮部
対応関係情報
対応関係情報
ファイル
変換後ファイルfb
変換元ファイルfa
ファイル入出力部
401
ファイル記憶部
31
30
ファイル内容
対応関係記憶部
34

FIG. 28

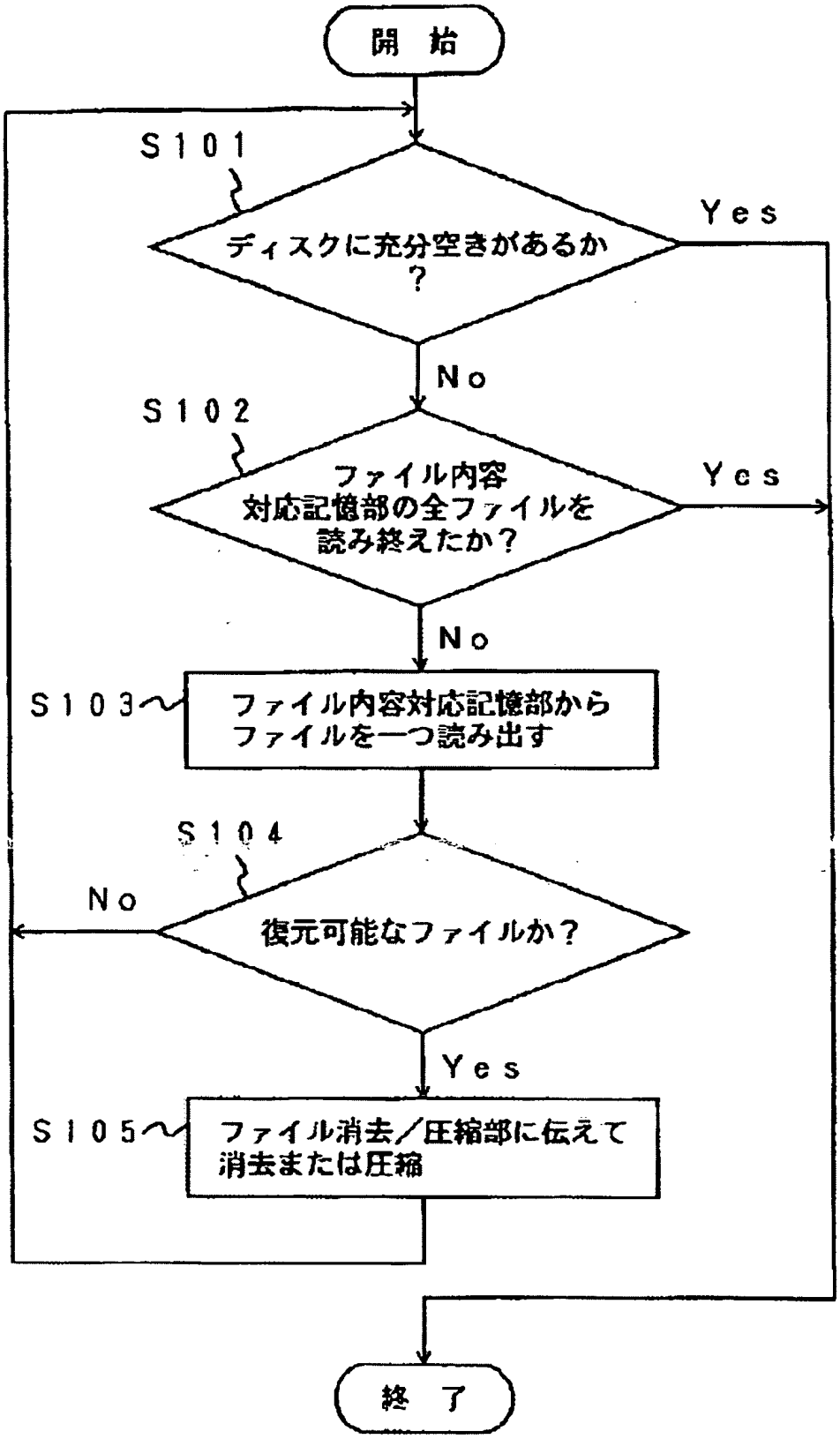

開 始
S101
ディスクに充分空きがあるか？
Yes
No
S102
ファイル内容対応記憶部の全ファイルを読み終えたか？
Yes
No
S103
ファイル内容対応記憶部からファイルを一つ読み出す
S104
No
復元可能なファイルか？
Yes
S105
ファイル消去／圧縮部に伝えて消去または圧縮
終 了

FIG. 29

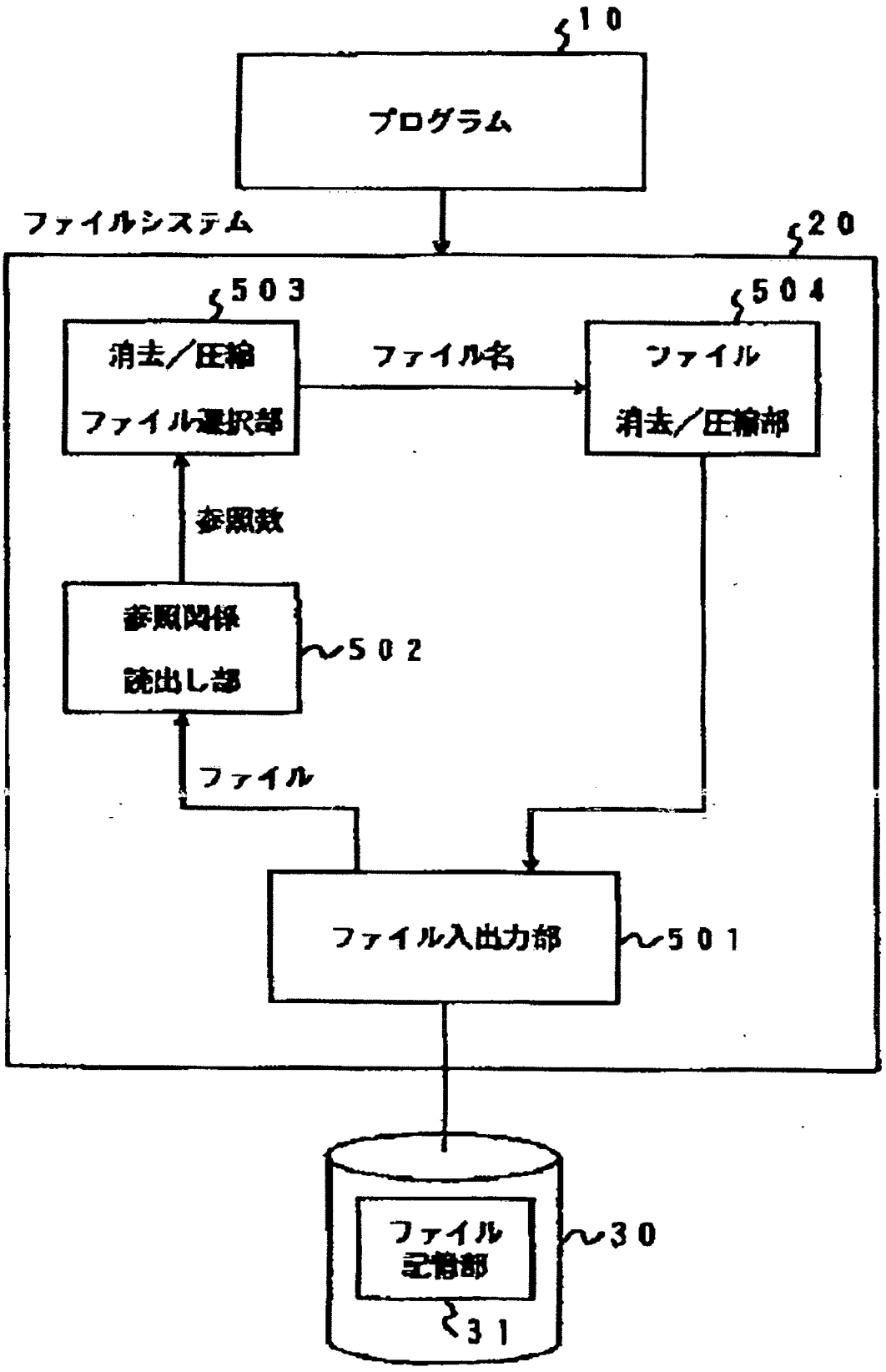

10
プログラム
ファイルシステム
20
503
504
消去/圧縮
ファイル選択部
ファイル名
ファイル
消去/圧縮部
参照数
参照関係
読出し部
502
ファイル
ファイル入出力部
501
ファイル
記憶部
30
31

FIG. 30

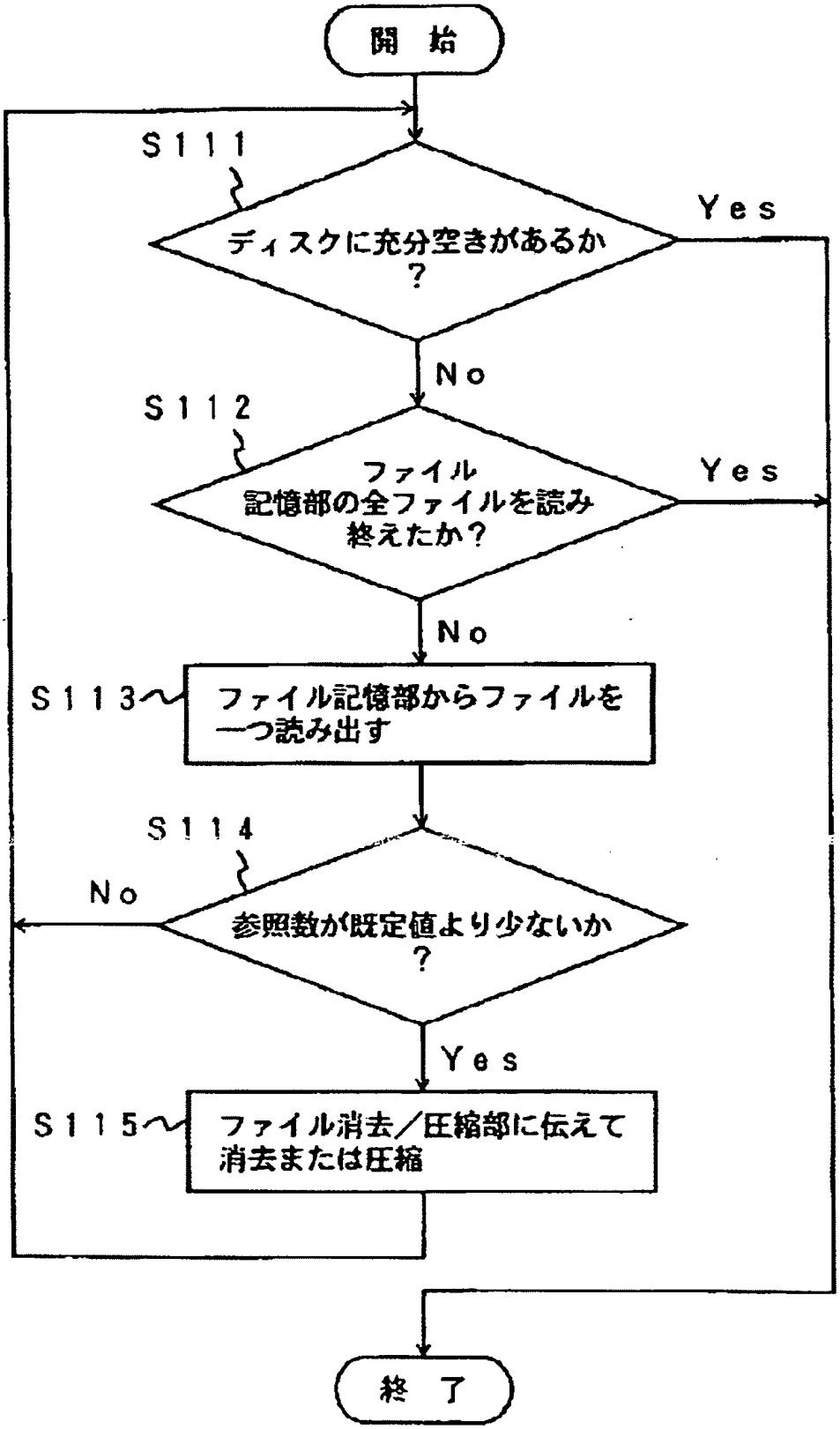

開 始
S111
ディスクに充分空きがあるか？
Yes
No
S112
ファイル記憶部の全ファイルを読み終えたか？
Yes
No
S113
ファイル記憶部からファイルを一つ読み出す
S114
No
参照数が既定値より少ないか？
Yes
S115
ファイル消去／圧縮部に伝えて消去または圧縮
終 了

FIG. 31

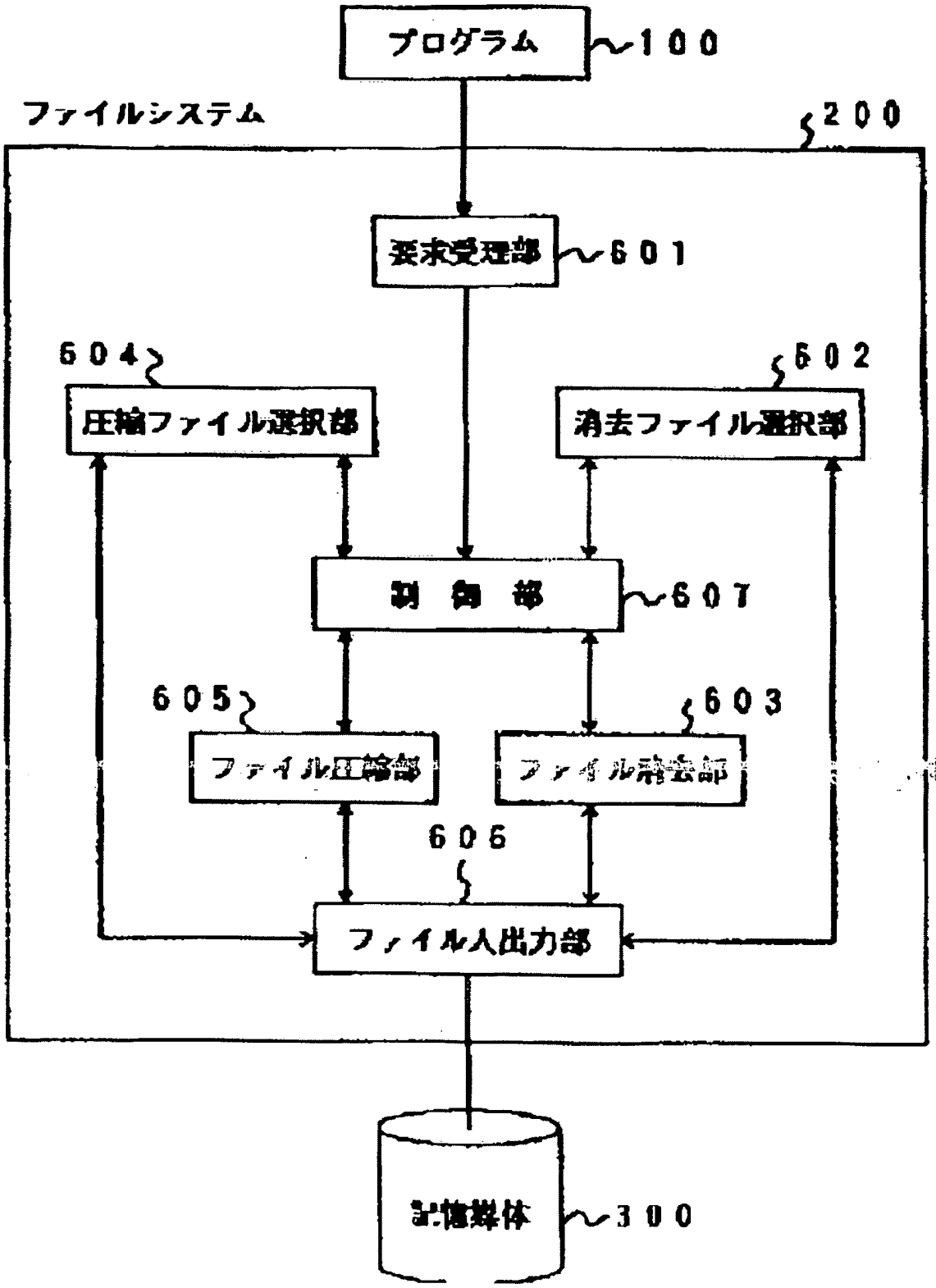

プログラム
100
ファイルシステム
200
要求受理部
601
604
圧縮ファイル選択部
602
消去ファイル選択部
制御部
607
605
ファイル圧縮部
603
ファイル消去部
606
ファイル入出力部
記憶媒体
300

FIG. 32

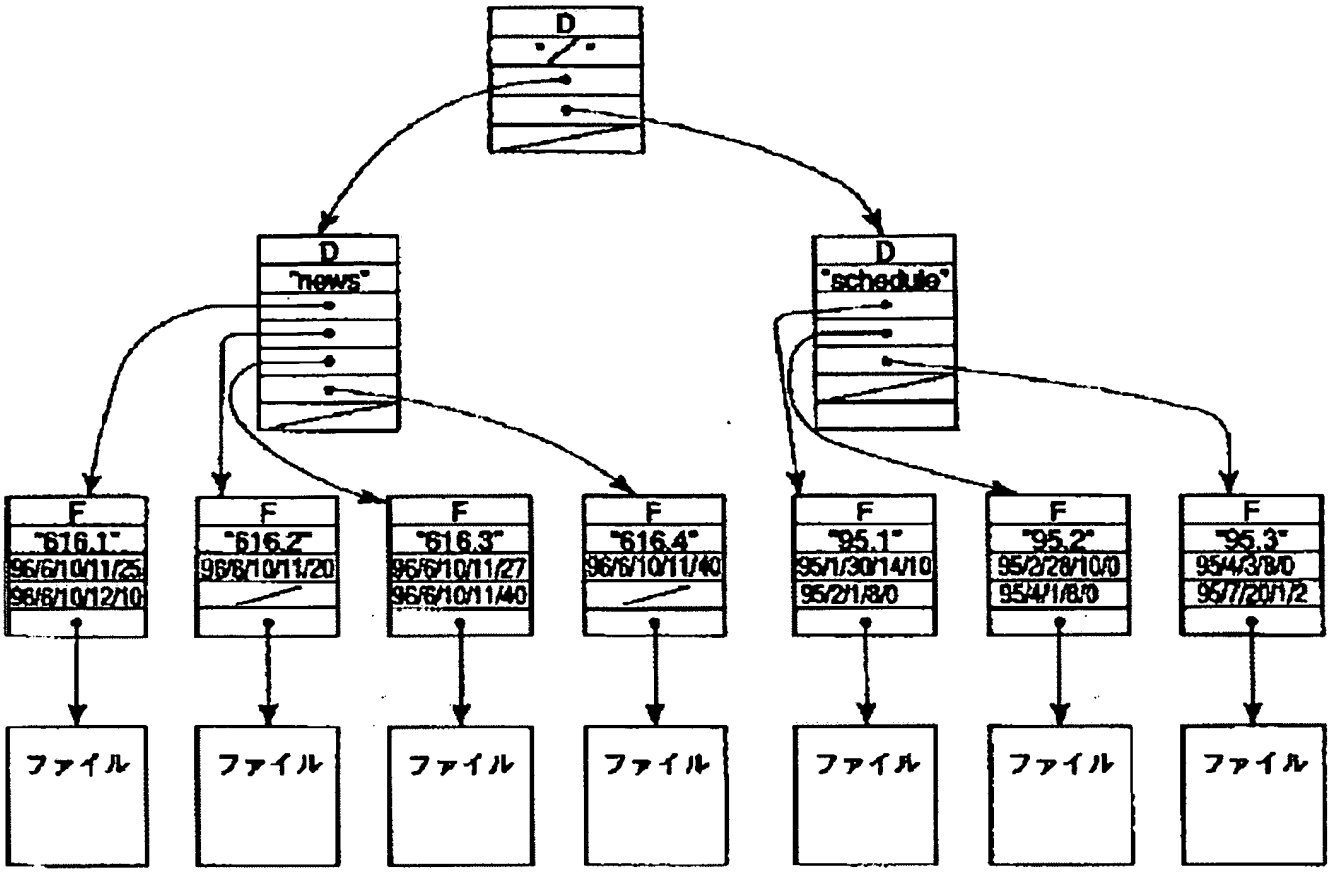

D
D
"news"
D
"schedule"
F
"616.1"
96/6/10/11/25
96/6/10/12/10
F
"616.2"
96/6/10/11/20
F
"616.3"
96/6/10/11/27
96/6/10/11/40
F
"616.4"
96/6/10/11/40
F
"95.1"
95/1/30/14/10
95/2/1/8/0
F
"95.2"
95/2/28/10/0
95/4/1/8/0
F
"95.3"
95/4/3/8/0
95/7/20/1/2
ファイル
ファイル
ファイル
ファイル
ファイル
ファイル
ファイル

FIG. 33

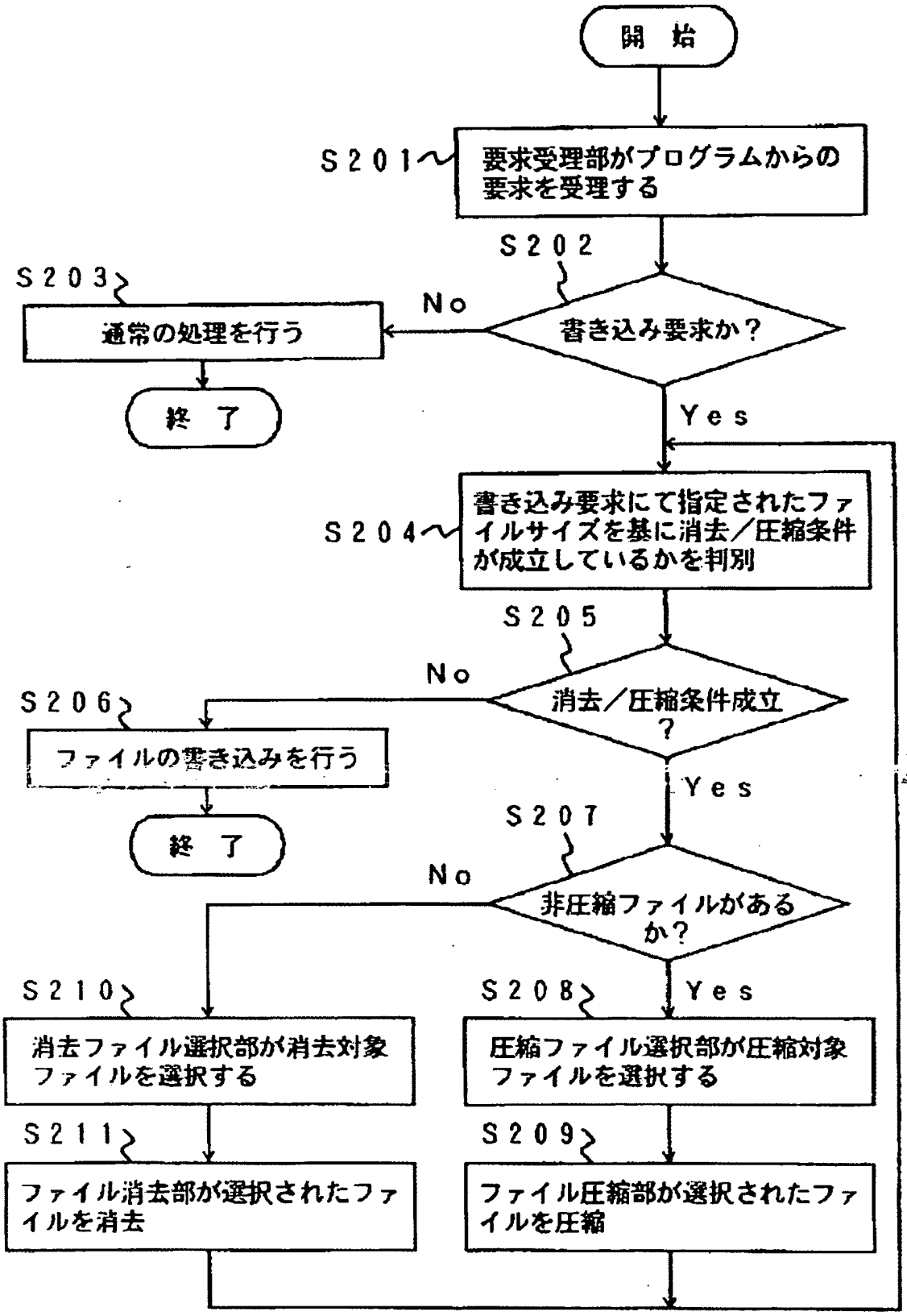
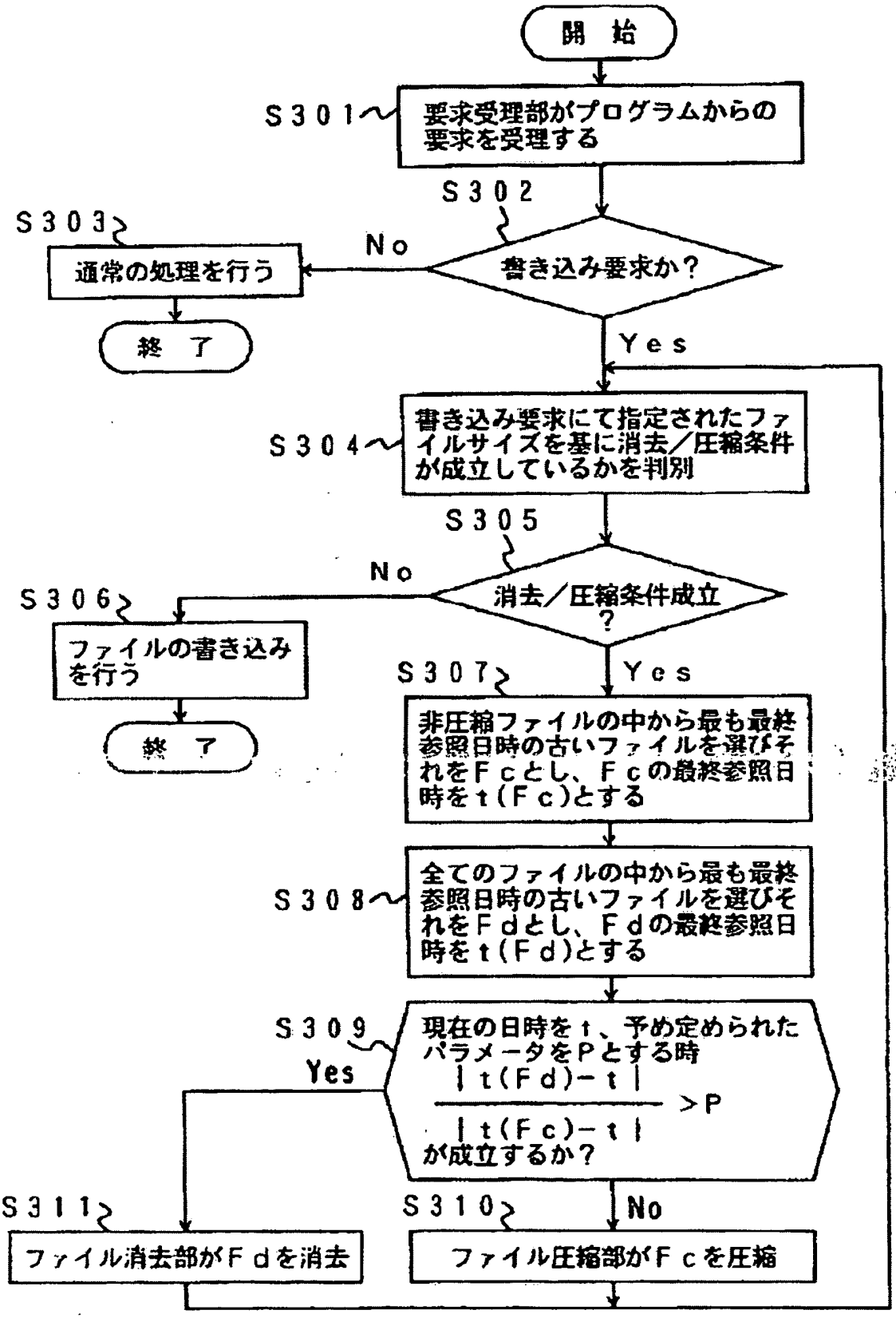

開始
S201
要求受理部がプログラムからの要求を受理する
S202
書き込み要求か？
No
S203
通常の処理を行う
終了
Yes
S204
書き込み要求にて指定されたファイルサイズを基に消去／圧縮条件が成立しているかを判別
S205
消去／圧縮条件成立？
No
S206
ファイルの書き込みを行う
終了
Yes
S207
非圧縮ファイルがあるか？
No
Yes
S210
消去ファイル選択部が消去対象ファイルを選択する
S208
圧縮ファイル選択部が圧縮対象ファイルを選択する
S211
ファイル消去部が選択されたファイルを消去
S209
ファイル圧縮部が選択されたファイルを圧縮

FIG. 34


開　始
S301
要求受理部がプログラムからの要求を受理する
S302
書き込み要求か？
No
S303
通常の処理を行う
終　了
Yes
S304
書き込み要求にて指定されたファイルサイズを基に消去／圧縮条件が成立しているかを判別
S305
消去／圧縮条件成立？
No
S306
ファイルの書き込みを行う
終　了
Yes
S307
非圧縮ファイルの中から最も最終参照日時の古いファイルを選びそれをＦｃとし、Ｆｃの最終参照日時をｔ(Ｆｃ)とする
S308
全てのファイルの中から最も最終参照日時の古いファイルを選びそれをＦｄとし、Ｆｄの最終参照日時をｔ(Ｆｄ)とする
S309
現在の日時をｔ、予め定められたパラメータをＰとする時
|t(Fd)−t| / |t(Fc)−t| > P
が成立するか？
Yes
S311
ファイル消去部がＦｄを消去
S310
No
ファイル圧縮部がＦｃを圧縮